Classifying imbalanced data sets using similarity based hierarchical decomposition

Classifying Imbalanced Data Sets Using Similarity Based Hierarchical Decomposition

Cigdem BEYAN (Corresponding author), Robert FISHER

School of Informatics, University of Edinburgh, G.12 Informatics Forum, 10 Crichton Street, Edinburgh EH8 9AB, UK

Tel: 44-131-651-3441 Fax: 44-131-650-6899

C.Beyan@sms.ed.ac.uk, rbf@inf.ed.ac.uk

Abstract

Classification of data is difficult if the data is imbalanced and classes are overlapping. In recent years, more research has started to focus on classification of imbalanced data since real world data is often skewed. Traditional methods are more successful with classifying the class that has the most samples (majority class) compared to the other classes (minority classes). For the classification of imbalanced data sets, different methods are available, although each has some advantages and shortcomings. In this study, we propose a new hierarchical decomposition method for imbalanced data sets which is different from previously proposed solutions to the class imbalance problem. Additionally, it does not require any data pre-processing step as many other solutions need. The new method is based on clustering and outlier detection. The hierarchy is constructed using the similarity of labeled data subsets at each level of the hierarchy with different levels being built by different data and feature subsets. Clustering is used to partition the data while outlier detection is utilized to detect minority class samples. The comparison of the proposed method with state of art the methods using 20 public imbalanced data sets and 181 synthetic data sets showed that the proposed method's classification performance is better than the state of art methods. It is especially successful if the minority class is sparser than the majority class. It has accurate performance even when classes have sub-varieties and minority and majority classes are overlapping. Moreover, its performance is also good when the class imbalance ratio is low, i.e. classes are more imbalanced.

Keywords: Class imbalance problem, Hierarchical decomposition, Clustering, Outlier detection, Minority-majority classes.

1. Introduction

In recent years, learning and classification with imbalanced data sets has become one of the key topics in pattern recognition due to its challenges especially for real-world applications where the data sets are dominated by normal examples in addition to a small amount of unusual examples [1, 2, 3]. Usually, the samples are grouped into binary classes. The well-represented class is called the majority class and the under-represented class is called the minority class. In such a case, a problem usually occurs because traditional classification algorithms tend to be biased towards to the majority class [4, 5]. On the other hand, even though being imbalanced is not always a problem (such as where the classes are separable) imbalanced data sets usually contain overlapping regions where the prior

probabilities of the two classes are almost equal [6]. Moreover, small disjuncts, and small sample size with high feature dimensionality [7] are frequently observed challenges in imbalanced data sets causing classification errors as well.

The appropriate evaluation criteria (such as the feature selection criterion to lead the training process and/or the criterion to evaluate the performance of classifiers) are also important issues when dealing with imbalanced data sets. For evaluation, many metrics exists in the literature. Accuracy is the most frequently used metric which is the sum of correctly predicted minority and majority samples over the total amount of samples. However, for imbalanced data sets, it is obvious that using accuracy might misguide the classifier and the importance of the minority class can be ignored since it is under-represented. This might be worse (total misclassification of the minority class) if the ratio between the classes is huge and the data is highly overlapping. Based on this, many alternative metrics has been proposed for evaluation of imbalanced classification. The geometric mean of sensitivity and specificity [8], adjusted geometric mean [9], Area Under Receiver Operating Characteristic curve (AUC) [10], and the F-measure which uses precision and recall (useful especially if we are highly interested in effective classification of a specific class) [5] are examples of effective metrics in this area.

Applications utilizing imbalanced data sets are diverse such as text categorization, medical diagnosis, fault detection, fraud detection, video surveillance, image annotations, anomaly detection [2, 4, 11]. Inherently, the diversity in applications has led to different solutions over the years. Approaches are traditionally divided into four categories: *i)* algorithmic level, *ii)* data level, *iii)* cost-sensitive methods and *iv)* ensembles of classifiers.

i) **The algorithmic level approaches** force the classifier to converge to a decision threshold biased to an accurate classification of the minority class such as by adjusting the weights for each class. For instance, in [3] a weighted Euclidean distance function was used to classify the samples using k-nearest neighbors (k-NN). Similarly, a Support Vector Machine (SVM) with a kernel function biased to the minority class is proposed in [12] to improve the minority class prediction.

ii) **The cost-sensitive approaches** assign different costs to training examples of the majority and the minority classes [13, 14]. However, it is difficult to set the cost properly (can be done in many ways) and may depend on the characteristics of the data sets. The standard public classification data sets do not contain the costs [2] and over-training is highly possible when searching to find the most appropriate cost.

iii) Re-sampling the data in order to handle the problems caused by the imbalanced nature of data is another approach. This **data level approach** does not modify the existing classifiers and is applied as a pre-processing technique prior to the training of a classifier. The data set can be re-sampled by oversampling the minority class [3], and/or under sampling the majority class [8, 15, 16]. Even though being independent of the classifier seems like an advantage, it is usually hard to determine the optimal re-sampling ratio automatically. Additionally, it might be problematic to oversample minority classes yet keep the distribution the same, especially in real-world applications where overlaps between minority and majority classes are highly likely. Similarly, while under-sampling the majority class, it is usually difficult to keep the new distribution of the majority class as similar as the distribution that it is sub-sampled from.

iv) **Ensembles of classifiers** have been popular in the last decade [17]. There are two main approaches; bagging and boosting. Bagging contains different classifiers which are applied to

subsets of the data [18]. Alternatively, in boosting, the whole set is used to train classifiers in each iteration while more attention is given to the classification of the samples that are misclassified in the previous iteration. This is done by adjusting the weights toward their correct classification. The most well known boosting method is AdaBoost [19]. Even though ensembles are frequently used for classification of imbalanced data sets, they are not able to handle the imbalanced data sets by themselves. And they require one or a combination of the approaches that are mentioned above such as re-sampling data (SMOTEBoost [20], EUSBoost [2] etc.).

In this study, we propose a new approach which is not completely defined by any of these categories presented above. The proposed method is a hierarchical decomposition which is based on clustering and uses outlier detection as the classifier. Following the standard approach in the literature, we consider only two-class problems with imbalanced data sets. The hierarchy is built using the similarity of data subsets while using the selected best feature subset (in terms of a chosen feature selection criterion) at each level of the hierarchy. Clustering of data based on the selected feature subset (without initially using known class labels) is the way to partition the data into separable subsets. Using outlier detection as the classifier is due to the assumption that the samples of the minority class are expected to be outliers and should be differentiated by the chosen outlier definition. For instance, outliers can be the samples that are far away from cluster center given a cluster composed of samples of the majority and the minority classes.

The basic steps of the proposed hierarchical decomposition are clustering (Section 3.1), outlier detection (Section 3.2), and feature selection (Section 3.3). The hierarchy is automatically generated using the similarities of data samples and does not use any class and/or feature taxonomy as hierarchical classifiers do. Many hierarchical classifiers are motivated by a taxonomy such as [21, 22]. In contrast to approaches which use the same feature space for all classifications we use different feature subsets at different levels of the hierarchy. This allows us to use more specific features once the data has become more focused onto specific subclasses (which might occur in the lower levels of the hierarchy).

The proposed method is evaluated using data sets from different fields and is compared with popular supervised learning methods in combination with algorithmic level and data level approaches. Additionally, synthetic data sets are used to test the performance of the proposed method in detail and different conditions (Section 4).

The contributions of this paper are as follows:

- We present a novel method that uses outlier detection in combination with clustering to classify imbalanced data sets,
- We present a new hierarchical decomposition method which does not use any fixed hierarchy based on features and/or classes. By being based on clustering, it is different from common hierarchical methods which use supervised learning,
- We show that different feature spaces can be used to build the hierarchy,
- Results show that the proposed method is successful especially when the distribution of the minority class is sparser than the majority class. It performs well when the class imbalanced ratio (the number of minority class samples over majority class samples) is low. It is successful if the majority and minority samples are highly overlapping and even when both classes contain varieties (such as having a mixture of distributions or having subclasses).

The paper is structured as follows: Section 2 briefly discusses the previous research on the class imbalance problem, hierarchical classifiers and hierarchical decomposition methods. Section 3 introduces the proposed method including the hierarchy construction (training step), new sample classification using the hierarchy (testing step) and basic methods (clustering, outlier detection, and feature selection). The test data sets, experimental set up and the corresponding results are given in Section 4. Finally, Section 5 discusses the proposed method with its advantages and shortcomings.

2. Related Research

The research related to the proposed method can be divided into two subsections: research considering the class imbalance problem and research about hierarchical classifiers and decomposition.

2.1 Class imbalance problem

One of the most popular *re-sampling approaches* is SMOTE [3] which creates synthetic minority class instances by pre-processing before classification. In this approach, for each minority class sample a new sample is created on the line joining it to the nearest minority class neighbor. Previously, it has been combined with many supervised methods such as SVM [5], Naive Bayes [3], C4.5 [2, 3], Random Forest [23, 24]. Even though it is popular and works better than only under-sampling the majority class, it does not always achieves better classification performance compared to the original classifiers as observed in [25]. A possible reason for this is that the newly generated samples might cause class overlapping. There are some improved versions of SMOTE as well such as Borderline SMOTE [26], SMOTEBoost [20], and modified SMOTE [26]. Recently, Barua et al. [27] proposed a novel oversampling method which identifies the most informative minority class samples, which are the samples hard to classify. Using clustering, weights were assigned to each minority samples according to their distance to majority class samples. Finally, the synthetic samples were generated using those weighted samples.

In addition to oversampling, *bagging* with oversampling for a bioinformatics application is presented in [28]. Differently, Liu et al. [29] proposes a double *ensemble classifier* by combining bagging and boosting. In that study, EasyEnsemble and BalanceCascade use bagging in the first ensemble and also for each bag AdaBoost [19] is used. Seiffert et al. [30] combined sampling and ensemble techniques (RUSBoost; random under sampling with boosting) to improve the classification performance in skewed data. The algorithm is close to SMOTEBoost [20] while being simpler, faster and performing much better. Random under sampling (RUS) randomly removes the majority class samples until the training set becomes balanced. However, the results shows that making the dataset completely balanced (imbalanced ratio=1) might result in lower performance than having imbalanced ratio a bit lower than 1. A *boosting algorithm* with an ensemble of SVM was presented in [31] where the minority class prediction is increased compared to pure SVM. In this study [31], Boosting-SVM with Asymmetric Cost is the best on average compared to very popular methods such as SMOTEBoost [20], random under sampling with SVM and SVM-SMOTE [3] for imbalanced data classification.

On the other hand, *cost sensitive approaches* which are usually applied in earlier works include many variations applied with k-nearest neighbors (kNN) [32], SVM [33], decision trees [34], logistic

regression [35] and so forth. For instance, imbalanced logistic regression (ILR) was advocated in [35] for mine classification which is based on logistic regression. The results show that ILR is better in classification of imbalanced datasets compared to pure logistic regression but not very successful on classification of datasets which have a high amount of outliers.

In summary, the number of proposed approaches in this field is very large. Interested readers can refer to review papers [17, 27, 36, 37, 38, 39] for a very detailed discussion of imbalanced data classification.

2.2 Hierarchical Classifiers and Hierarchical Decomposition

Hierarchical methods for classification can be considered in two categories: *i)* Hierarchical classifiers, *ii)* Hierarchical decomposition. In the former case, there is a pre-defined hierarchy such as a taxonomy and the classes are organized using this taxonomy as a tree or a graph. In the latter case, there is no pre-defined class hierarchy and the hierarchy is created using factors such as similarity of data [40].

Many studies [41, 42, 43, 44] have focused on hierarchical classifiers. Wu et al. [42] used a tree shaped class taxonomy for a multi class scenario while using a multi-class classifier at each parent node. Li et al. [43] presented a method for automatic music genre classification. In that paper [43], the taxonomy gives the relationship between the genres and in addition to genre classification automatic taxonomies were built by using the similarity matrix from linear discrimination. Classification in large taxonomies was re-visited with improved results in [44]. In that work [44], solutions for error propagation (which affects the classification of the lower levels of the hierarchy much more) and the complex decision boundaries occurring in higher levels of the hierarchy were studied. Silla et al. [45] proposed a method based on a fixed taxonomy for hierarchical protein function prediction. Given the fixed taxonomy, a couple of strategies were applied: selecting best classifier, selecting the best feature representation given a fixed classifier and selecting the best classifier and best feature representation.

Hierarchical decomposition is as popular as hierarchical classifiers and in this study we are also proposing a method for this category. The most common type of hierarchical decomposition is dividing a multi class problem in a hierarchical way to obtain binary hierarchical classifier [40]. In this technique, a hierarchy can be created using the similarity of classes. For instance, Kumar et al. [46] divided classes in a hierarchical way where classes similar to each other are grouped together. This turns the multi-class classification problem into a binary classification problem. Similarly, hierarchical SVM was presented in [22] for multi-class classification. In that method, classes are partitioned into two subsets until one class label is obtained at a leaf node based on class similarities. SVM based hierarchical clustering was used for text mining utilizing the similarities between features [47]. Dividing the problem into smaller problems by the hierarchy results in selecting a smaller set of features (a more specific domain term features) to a sub-problem which increased the accuracy and efficiency [47]. Freitas et al. [48] proposed a method for generation of meta-classes on the fly without using a taxonomy. In this work [48], a two level hierarchy was constructed where the leaf is composed by the similarity of the meta-class level. Epshtein and Ullman [21] built an automatic hierarchy using the relationship between features. The same feature extraction procedure is applied at all levels of the hierarchy. The top-level features are broken into their smaller components and for all levels of the hierarchy different features, sub-features and their specific parameters are learned using the training samples. The results showed that dividing features into a hierarchy performs better than using features

as a whole. Hierarchical clustering is combined with entropy based feature selection to construct a binary hierarchical structure [49]. The method results in a hierarchy which has different feature subsets. At each component of the hierarchy SVM is used as the classifier. This method [49] performed better than traditional one-against-one hierarchical decomposition for multi-class audio event classification for healthcare applications.

Studies such as [21, 47, 50, 51] showed that hierarchical methods can have better classification performance compared to flat classification techniques. More interested readers can refer to [40] for a recent survey which combines different hierarchical classifiers in different application areas and highlights the differences between hierarchical classifiers and hierarchical decomposition.

In this study, the proposed hierarchy is not based on any taxonomy between features or classes. It is a hierarchical decomposition technique which uses feature selection to find the similarities between data and an outlier detection method to determine the data samples belong to a specific level of the hierarchy. Therefore, the different levels of hierarchy use different data and feature subsets. Additionally, the class imbalance problem that we consider is a two class problem embedded in a hierarchy.

3. Hierarchical Decomposition for Imbalanced Data Set Classification

The basic components of the proposed hierarchical method are *i)* clustering (Section 3.1), *ii)* outlier detection (Section 3.2) and *iii)* feature selection (Section 3.3). Feature selection is embedded into clustering and outlier detection. Clustering of data on selected features without using data labels partitions the data into clusters some of which might be separable. In Section 3.4, the training step of the proposed method, which is the construction of the hierarchy, is given. To automatically generate the hierarchy, data is first clustered then classified by outlier detection using the ground truth data. In contrast to previous research that uses the same feature set for every level of the hierarchy or a flat classifier, we use different feature sets at different levels of the hierarchy (similar to [45] while our method is different by not using a fixed taxonomy). As the last part of this section (Section 3.5) we describe the classification phase of the proposed method which is called as "new data sample classification using the hierarchy".

3.1 Clustering

In this study, we used Affinity Propagation (AP) [52] for clustering. AP has been applied as a clustering method in various studies including anomaly detection which is an imbalanced data set application. AP identifies the cluster centers from actual data points which are called *cluster exemplars*. The method uses the pair-wise similarity of each pair of feature points which is the negative of the Euclidean distance between the points. The objective function of AP tries to find the exemplars that maximize the overall sum of similarities between all exemplars and their data points given the similarity matrix. There are two kinds of messages between data points. The first message, which is called *responsibility,* is from data point i to j that represents the accumulated evidence for how appropriate it would be for the data point j to be the exemplar for data point i. The second message, which is *availability*, represents how appropriate it would be for data point i to choose data point j as its exemplar. More information can be found in [52].

AP has many advantages over traditional clustering methods such as its fast processing speed, being non-parametric, not requiring initialization, not depending on sample order (such as hierarchical clustering) and scalability (which makes our method scalable as well). However, in our case the main reasons for using this method are its ability *i)* to produce smaller clusters, and *ii)* to produce uneven sized clusters which is compatible with the outlier detection method that we used.

3.2 Outlier Detection

An outlier is defined as a datum which is distant from other data points in the same cluster. Most of the time, the cardinality of the outliers is smaller than the other data points in the same cluster. By using this definition in an imbalanced data set problem outliers become samples of the minority class.

In this study, we adapted the outlier detection method from [53] and use it to detect minority class samples. This is the foundation of classification in the proposed method. We assume two types of outliers (Figure 1):

- Those located in small clusters,
- Those in dense clusters but far from center of the cluster.

To detect the small and dense clusters, a threshold is defined based on the cardinality of all clusters. A cluster which has fewer data samples than *10%* of the median cardinality of clusters or a cluster that has only one data point is defined as a small cluster. All samples that belong to such a cluster are classified as the minority class (Figure 1a, the clusters having boundaries with thick lines). Otherwise, the cluster is a dense cluster, and outliers are detected using the Euclidean distance between the sample and the cluster exemplar (Figure 1a, the clusters having boundary with dashed lines and Figure 1b). A data sample whose distance is longer than the threshold $\tau=\mu+w\sigma$ (with mean (μ), weight (w) and standard deviation (σ) of all distances between all samples and cluster exemplar) of that cluster is defined as an outlier and this makes it belong to the minority class (Figure 1b). This threshold is different and specific for a given cluster and is calculated in terms of the data in its cluster. The *w* is taken as *{-1, -0.3, 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.9, 1, 1.5, 2, 2.5, 3}* in the experiments given in Section 4. For interested readers, evolutionary algorithms can be adapted to find the optimal *w* but in our experiments, the values of *w* that we used were good enough to obtain good performances.

(a) (b)

Fig. 1: (a) A representation of clustered data, for small clusters boundaries are shown with thick lines and dense clusters' boundaries are shown with dashed lines, (b) Outlier detection in dense clusters: samples which are inside of the inner circle are classified as the majority class whereas the rest of the samples are classified as the minority class, given threshold τ.

3.3 Feature Selection

As mentioned before, feature selection is embedded in clustering and outlier detection. In this step, we use Sequential Forward Feature Selection [54] to find the best feature subset. Feature selection helps to decrease the chance of over-fitting, eliminate irrelevant, redundant features and even features that might misguide clustering. Different from the standard procedure of Sequential Forward Feature Selection, we use the mean of sensitivity (Eq. 1) and specificity (Eq. 2) (as suggested in [8] for imbalanced data set evaluation) as the feature selection evaluation criteria rather than the accuracy which increases misclassification of the minority class especially when the number of minority class samples are very low.

Table 1: Contingency table for two-class problems.

	Prediction as positive class *(Minority Class)*	Prediction as negative class *(Majority Class)*
Positive Class *(Minority Class)*	True Positive (TP)	False Negative (FN)
Negative Class *(Majority Class)*	False Positive (FP)	True Negative (TN)

$$\textit{Sensitivity (True Positive Rate)} = TPrate = TP/(TP+FN);\ \textit{minority class accuracy} \quad (1)$$

$$\textit{Specificity (True Negative Rate)} = TNrate = TN/(TN+FP);\ \textit{majority class accuracy} \quad (2)$$

Feature selection is used as follows: Given the current set of features, an additional feature is added, clustering and outlier detection are performed with the extended feature set. The mean of sensitivity and specificity are found using the ground-truth labels. All possible additional features are tried in the same way, and the extension with the best classification performance using the training set is kept. Adding features to the current feature subset stops when the classification performance on the training set decreases compared to the previous feature subset.

The experiments were evaluated using different feature selection algorithms as well. We applied Laplacian score [55] and Multi-cluster approach [56] that are a filtering based feature selection algorithms (meaning that class labels are not being used). The results showed that Sequential Forward Feature Selection performs better than these filtering methods even though it is much slower. We tried different feature selection criteria as well, such as precision and recall, accuracy, mutual information etc. using the geometric mean of sensitivity and specificity as the evaluation metric. Those criteria did not perform as well as the mean of sensitivity and specificity.

3.4 Hierarchy Construction

We start with two definitions:

- *Perfectly classified clusters*: Contain the samples that are all correctly classified by the outlier detection process. A perfectly classified cluster can include:
  - Both minority and majority class samples which are correctly classified using the outlier detection threshold (*perfectly classified mixed*, Figure 2a),
  - Only majority class samples which are correctly classified using the outlier detection threshold (*perfectly classified pure majority*, Figure 2b)
  - Only minority class samples which are correctly classified due to being in a small clusters where we assume that samples of small clusters are outliers (*perfectly classified pure minority*, Figure 2c)

- *Misclassified clusters:* Consist of at least one data sample that is not correctly classified by the outlier detection process. A misclassified cluster can contain:
  - Both minority and majority class samples with at least one sample wrongly classified using the outlier detection threshold (*misclassified mixed*, Figure 2d),
  - Only majority class samples with at least one sample classified as a minority sample using the outlier detection threshold (*misclassified pure majority when the cluster is a dense cluster*, Figure 2e),
  - Only majority class samples which are wrongly classified as minority samples due to being in a small cluster (*misclassified pure majority when the cluster is a small cluster*, Figure 2f),
  - Only minority class samples where at least one sample is classified as a majority sample using the outlier detection threshold (*misclassified pure minority*, Figure 2g).

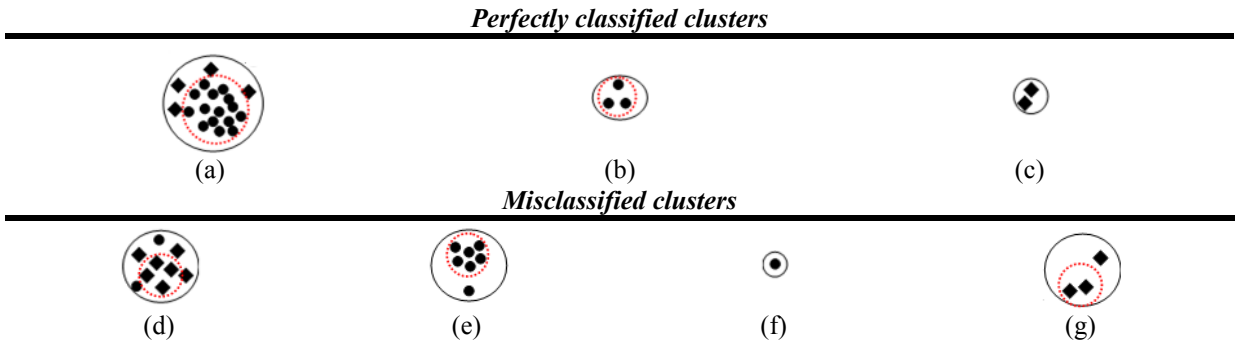


Fig. 2: Definitions used in hierarchy construction: perfectly classified clusters, misclassified clusters. A perfectly classified cluster can be a) perfectly classified mixed, b) perfectly classified pure majority, c) perfectly classified pure minority. A misclassified cluster can be d) misclassified mixed, e) misclassified pure majority when the cluster is a dense cluster, f) misclassified pure majority when the cluster is a small cluster, g) misclassified pure minority. Diamonds represent the samples from the minority class while circles represent the samples from the majority class. The outlier detection thresholds for dense clusters are shown with dashed red circle.

The training phase constructs a hierarchy which involves the selection of features and determine the clusters with their outlier detection threshold (which is specific for each cluster and is only applied to dense clusters) while a subset of data is used at each level of the hierarchy. All of the training data

points are used for the construction of the first level of the hierarchy. Any data which belongs to a perfectly classified cluster at any level are not used at further levels.

At each level of the hierarchy, a subset of features, which are found by feature selection (Section 3.3) are used for clustering. Following this, outlier detection is applied to each cluster (Section 3.2) and the minority class and majority class predictions are determined. Then, using the ground truth labels the sensitivity (Eq. 1) and specificity (Eq. 2) are found. Feature selection stops when the value of the feature selection criterion on the training set (for the specific level) decreases compared to the previous iteration of feature selection. This determines the best feature set for the current level. After the best feature set is found, clusters are labeled either as *perfectly classified clusters* or *misclassified clusters* using the ground truth data. The figure illustrating those steps is given in Figure 3.
All *perfectly classified clusters* are fixed for that level of the hierarchy and hierarchy building recurs with the samples which belong to *misclassified clusters* (all of these samples are collected together for the next level of feature section, clustering and outlier detection). The tree is extended recursively until there is no perfectly classified cluster or every sample is perfectly classified. The leaf nodes of the hierarchy are either *i)* perfectly classified clusters which can be observed mostly at the upper levels, *ii)* misclassified clusters which can only be at the bottom level of the hierarchy.

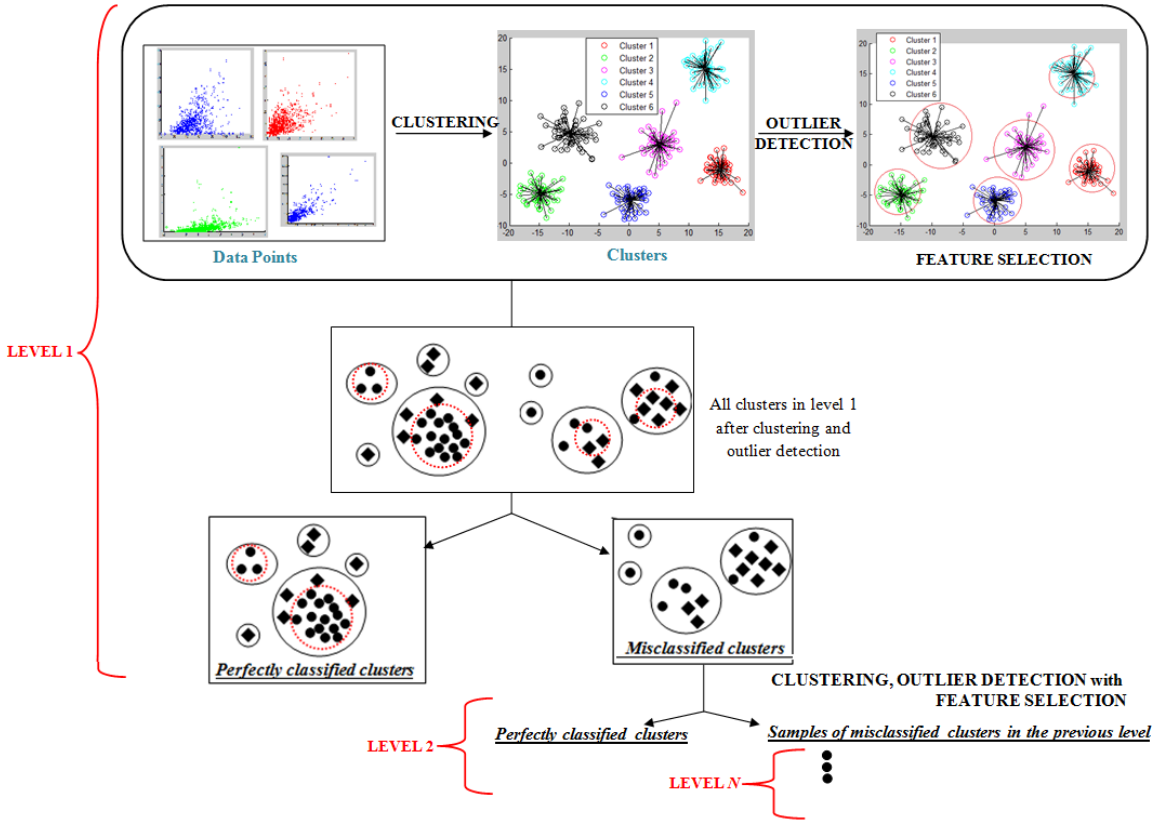


Fig. 3: (a) Overview of proposed method: Diamonds represent the samples from the minority class while circles represent the samples from the majority class. The outlier detection thresholds for dense clusters are shown with dashed red circle. After clustering and outlier detection, clusters are labeled either as perfectly classified clusters or misclassified clusters using the ground truth data. Perfectly classified clusters contain the samples all correctly classified by the outlier detection and decomposition for the samples belong to perfectly classified clusters stops at this point. Misclassified clusters consist of at least one sample that is not correctly classified. Clustering, feature selection and outlier detection is repeated for the samples of misclassified clusters.

The pseudo-code for training is:

```
Input: Training Set: X ={X_1, X_2,...X_N}
       Ground-truth labels: G ={G_1, G_2,...G_N}
       Size of training set: N
       Features: F ={f_1, f_2,...,f_M} % all possible features
       Total number of features: M
       Feature Selection Criterion Function: E
       Outlier detection thresholds: w={-1, -0.3, 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.9, 1, 1.5, 2, 2.5,3}
Output: Hierarchy H ={ Total number of levels: L
                       Selected Feature Subsets: selFea ={sf_1, sf_2,... sf_L}, where sf_i ⊂ F
                       Perfectly Classified Clusters: C_perfect ={C_P1, C_P2,... C_PL}
                       Misclassified Clusters: C_mis ={C_M1, C_M2,...C_ML}
                                          where C_Pi and C_Mi ⊂ set of all subsets of X
begin:
   for z=1:size(w)
      w_z=w(z); % current outlier detection threshold
      current_level=1
      while current_level >=1
            if current_level ==1
                  remaining_samples=X;
            else
                  remaining_samples=samples(C_Mcurrent_level-1);
            end
            featureSelection_converged=false;
            sf_current_level={}; F̂=F;    % all features
            while (NOT featureSelection_converged)
                  for f_i ∈ F̂
                      [C] =Clustering (remaining_samples, (sf_current_level ∪ {f_i}));
                                                      % using algorithm in Section 3.1
                      [C_Pi, C_Mi] =OutlierDetection (C, w_z);    % using algorithm in Section 3.2
                    e_i =evaluate( C_Pi, C_Mi, G);
                  end
                  select j=argmax_i ei
                  sf_current_level= sf_current_level ∪ {f_j}
                  F̂ =F̂ \ {f_j}
                  featureSelection_converged =E(sf_current_level) ≤ E(sf_current_level \ {f_j})
            end
            H. L =current_level;
            H. C_perfect(H.L) =C_Pj-1;
            H. C_mis(H.L) =C_Mj-1;
            H. selFea(H.L) =sf_current_level\{f_j};

            if notEmpty(H.C_perfect(H.L)) and size(samples(H.C_perfect)) ≠ N
                                          % there is at least one perfectly classified cluster
                                          % and the total number of perfectly classified
                                          % samples are not equal to N
                current_level =current_level+1;
            else
                 current_level =0;
            end
       end
     end
end
```

3.5 New Data Sample Classification Using the Hierarchy

During the new data sample classification (testing), the constructed hierarchy including the misclassified clusters of each level, the selected feature subset for each level and the outlier detection threshold for each cluster are used. Testing is rule based and very efficient since it is based on distance calculations between the new data point and the clusters in each level to find the closest cluster. The closest cluster is found using the Euclidean distance between the test data sample's feature vector (using the features selected for the current hierarchy level) and all cluster centers at a given level.

At each level in the hierarchy, *the closest cluster* can be one of 6 possible cluster types ("*perfectly classified pure minority*", "*perfectly classified pure majority*", "*perfectly classified mixed*", "*misclassified pure majority*", "*misclassified pure minority*", and "*misclassified mixed*") as described in Section 3.4. At each level in the hierarchy, for the new data sample, 3 types of class decisions are possible: "***majority class sample***", "***candidate minority sample***" and "***no effect on the decision***" Figure 4 summarizes the testing algorithm.

At the given hierarchy level (with its trained clusters, outlier detection thresholds, and selected features) and a new data sample, the first step of the algorithm is to compute the characteristics of the closest cluster, which can be one of the followings:

i. The closest cluster is a "perfectly classified pure minority cluster" (underlined with red and shown as "a)" in Figure 4) which makes the new data sample a "***candidate minority sample***". The new data sample goes to the next hierarchy level.
ii. The closest cluster is a "perfectly classified pure majority cluster" (underlined with red and shown as "b)" in Figure 4) and the new data sample is further than the outlier detection threshold of that cluster. This makes the new data sample a "***candidate minority sample***". The new data sample goes to the new hierarchy level.
iii. The closest cluster is a "perfectly classified pure majority cluster" and the distance between the new data sample and the corresponding cluster's centre is smaller than the outlier detection threshold of that cluster. This makes the new data sample a "***majority class sample***" and classification stops.
iv. The closest cluster is a "perfectly classified mixed cluster" (underlined with red and shown as "c)" in Figure 4) and the new data sample is further than the outlier detection threshold of that cluster which makes the new data sample a "***candidate minority sample***". The new data sample goes to the next hierarchy level.
v. If case *iv* occurs, but the distance between the new data sample and cluster centre is smaller than the threshold, then the new data sample is a "***majority class sample***" and classification stops.
vi. The closest cluster is a "misclassified cluster" (pure or mixed) (underlined with red and shown as "d)" in Figure 4) then the data sample proceeds to the next level. This does not have any effect on the classification of the new data sample unless all the closest clusters at each level are "misclassified cluster" ("***no effect on the decision***", see below for the applied rules in this case).

As seen, even a single level's decision of the majority class is enough to classify the new data sample as "majority sample" regardless of the level of the hierarchy. On the other hand, if there is no decision as ***majority class sample*** from any level and if the decision of at least one level is "***candidate minority sample***" (underlined with green and shown as "a)" In Figure 4) then the class of the new sample is declared to be "minority sample". However, as mentioned in case *vi*, it is possible that the closest cluster at each level of the hierarchy is a misclassified cluster (underlined with green and shown as

"b)" in Figure 4). In this case, we use the ground-truth labels of the training samples and apply the following rules, starting from the top of the hierarchy:

vii. The closest cluster at the current level contains all majority class training samples by looking at the ground-truth class labels: If the new data sample is not further than the rest of the samples in that cluster this makes it a "***majority class sample***" (classification stops here); otherwise the data goes to the next hierarchy level (underlined with purple and shown as "b)" in Figure 4).

viii. The closest cluster contains all minority class training samples by the ground-truth: The data is sent to the next hierarchy level (underlined with purple and shown as "a)" in Figure 4), to apply the rules *vii, viii, ix,* and *x* again.

ix. The closest cluster contains both majority and minority training samples: In this case, we apply the nearest neighbor rule (underlined with purple and shown as "c)" in Figure 4) which makes the class of the new sample the same as the closest training sample's class. If the class is majority class then classification stops. Otherwise, the data goes to the next level to apply the rules *vii, viii, ix,* and *x*.

x. If the data reaches the last level and were not classified as majority class sample in the last level as well, then the data is classified as a "***minority class sample***".

In this study, we used the heuristic that a decision as a "***majority class sample***" at any level stops the classification of the new sample while a decision as a "***minority class sample***" sends the new data sample to the next hierarchy level. Other strategies can also be applied. For instance, for applications where the classification of the minority class is more important than classification of majority class (such as anomaly detection), the heuristics can be applied conversely. Then, any decision as a "***minority class sample***" stops the classification regardless of the level of the hierarchy while decision as a "***candidate majority sample***" sends the new sample to the next hierarchy level (see Section 5 for more detailed discussion). Alternatively, all the samples can traverse all levels of the hierarchy while decisions are made as "***candidate majority class***" and "***candidate minority class***". In this case, the final class decision can be done by majority voting.

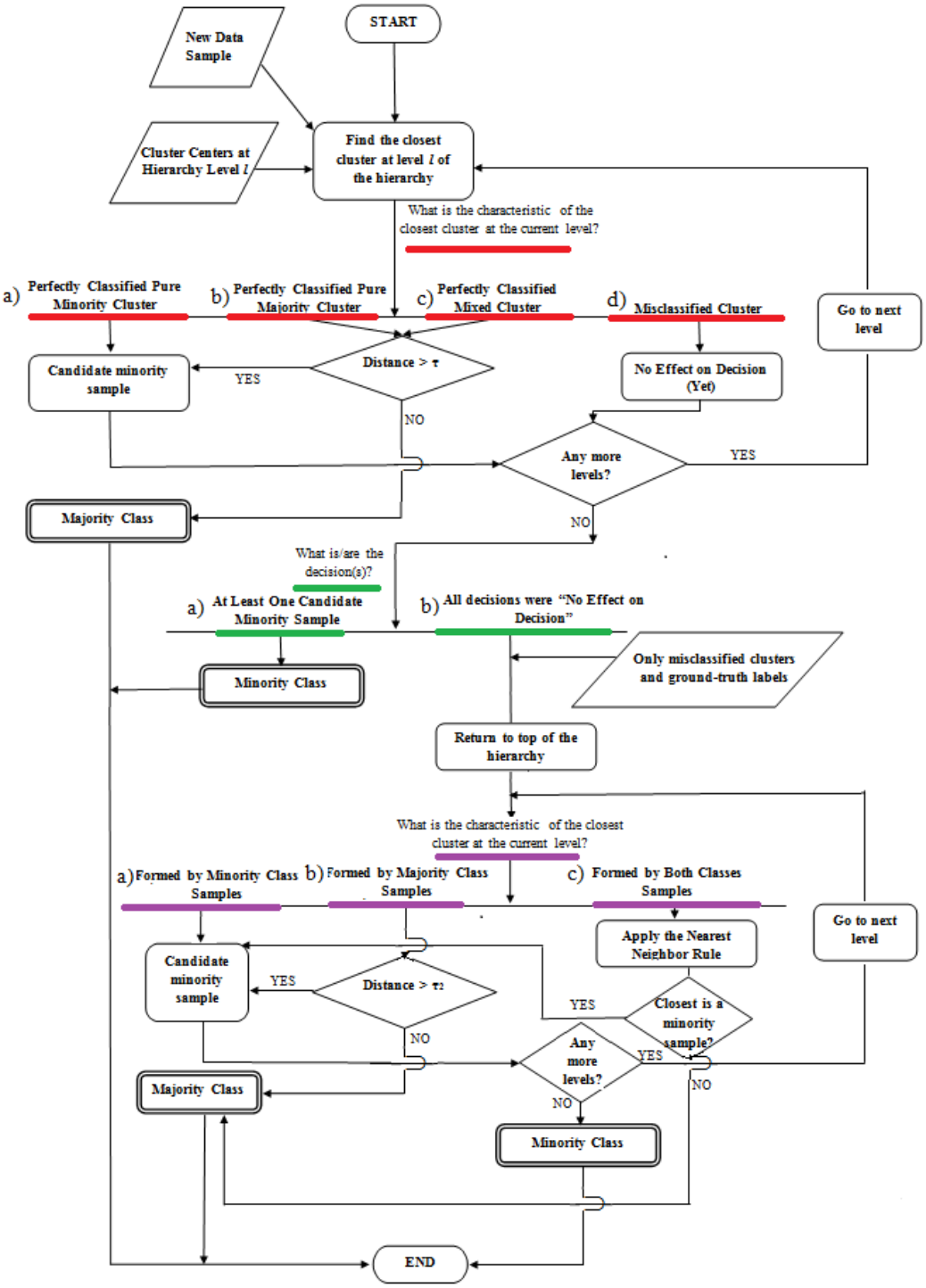


Fig. 4: The flow chart of classification of a new data sample using a previously constructed (during training) hierarchy. The characteristic of the closest cluster can be: *a)* perfectly classified pure minority, *b)* perfectly classified pure majority, *c)* perfectly classified mixed, *iv)* misclassified cluster (either pure minority, pure majority or mixed) (those are all underlined with red). The decisions can be: *a)* majority class, *b)* minority class, *c)* candidate minority sample and *iv)* no effect on decision (which needs another iteration to classify the new sample's class as majority class or minority class). Decisions are all shown with rounded rectangles either with single or double line. Rounded rectangles with double lines represent the final class of the new sample whereas single line rounded rectangles indicate provisional decisions.

4. Experiments and Results

To evaluate the classification performance of the proposed method, the experimental setup can be divided into two sections: *i)* experiments using public imbalanced data sets and *ii)* experiments with synthetic data sets. In both parts the preprocessing algorithms that are given in Table 2 are applied.

Table 2: Preprocessing algorithms that are used.

Method	Description
Feature Selection [54]	Sequential forward feature selection method with the criterion of the mean of sensitivity and specificity was used as described in Section 3.3.
SMOTE [3]	Number of neighbors were selected as to make the dataset's imbalance ratio (the number of minority class samples over majority class samples [2, 4]) equal to 1. If this was not possible (when imbalance ratio is too small), then we took the number of neighbors equal to the number of minority class samples which made the set as balanced as it can be.
Under Bagging (Balanced Training [57])	This was only applied with Random Forest. All minority samples were kept, and subsets of the majority class were chosen randomly to build the decision trees. The number of majority class examples in the chosen subset was equal to the number of total minority class data samples.

4.1 Experiments with Public Imbalanced Data Sets

In this section, we introduce the data sets used and previous state of the art classification algorithms to compare their performance with the proposed method. The results are evaluated in terms of different metrics. Moreover, different statistical tests were applied to assess the performance significance between the proposed method and the state-of-art methods.

4.1.1 Data Sets

Twenty popular imbalanced data sets were used to evaluate the effectiveness of the proposed method. The data sets are from different fields such as biology, physics, medicine, etc. The number of features (#Fea.), the total number of samples (#Sam.), the total number of minority and majority samples (#Min., #Maj.), the imbalance ratio (IR=the number of minority class samples over majority class samples [2, 4]) and the corresponding citations for each data set (Ref.) are given in Table 3. While choosing these data sets, we tried to cover the range of variety in the data sets. The selection was based on: unique datasets name (as many of the data sets are combinations of the same data set but with different class combinations), a range of IR measure values (from 0.57 to 0.02), variation in the amount of class overlap (see the KEEL repository [58] for more information), a varying number of samples (from 106 to 7420) and variation in the number of features (from 7 to 294).

Table 3: Summary of imbalanced data sets.

Data Sets	#Fea.	#Sam.	(#Min., #Maj.)	IR	Ref.
Ionosphere	34	351	(126, 225)	~ 0.57	[59, 60]
Pima	8	768	(268, 500)	~ 0.50	[61, 62]
Vehicle1	18	4230	(1085, 3145)	~ 0.35	[58]
Vehicle2	18	4230	(1090, 3140)	~ 0.35	[58]
Vehicle0	18	4230	(995, 3235)	~ 0.31	[58]
Hepato	9	536	(116, 420)	~ 0.28	[61]
Appendicitis	7	106	(21, 85)	~ 0.25	[63]
Satimage	36	6435	(626, 5809)	~ 0.11	[60]
Glass2	9	1070	(85, 985)	~ 0.09	[58]
Ecoli-0-1-4-7_vs_2-3-5-6	7	1680	(145, 1535)	~ 0.09	[58]
Ecoli-0-1-4-7_vs_5-6	6	1660	(125, 1535)	~ 0.08	[58]
Cleveland-0_vs_4	13	865	(65, 800)	~ 0.08	[58]
Scene	294	2407	(177, 2230)	~ 0.08	[64]
Yeast-1_vs_7	7	2295	(150, 2145)	~ 0.07	[58]
Ecoli4	7	1680	(100, 1580)	~ 0.06	[58]
Oil	49	937	(41, 896)	~ 0.05	[65]
Glass5	9	1070	(45, 1025)	~ 0.04	[58]
Yeast5	8	7420	(220, 7200)	~ 0.03	[58]
Yeast-1-2-8-9_vs_7	8	4735	(150, 4585)	~ 0.03	[58]
Winequality-red-8_vs_6-7	11	4275	(90, 4185)	~ 0.02	[58]

The Hepato data set originally had 4 classes, the Scene data set originally had 6 classes, Satimage data set originally had 7 classes. For those data sets, we chose the smallest class as the minority class and collapsed the rest of the classes into one in order to obtain a two-class imbalanced data set. The other data sets (Pima, Ionosphere, Appendicitis and data sets from the KEEL repository [58]) originally had binary classes or they were supplied as binary by the given references therefore we used those data sets as they are provided.

4.1.2 Results

To evaluate the proposed method 2 fold cross validation with the Appendicitis data set and 5 fold cross validation for the rest of the data sets was performed. The datasets from the KEEL repository [58] were provided as 5-fold. We used the testing sets of the corresponding data sets as provided but to obtain the validation set (which we need for feature selection especially) we randomly divided the supplied training sets into 4 folds where minority and majority class samples were distributed equally. This gave us datasets having equal amounts of samples for testing and validation with 3 times bigger training sets (as 5 fold cross validation with training, validation and testing set gives). Similarly, for the rest of the data sets, using 5 fold cross validation, training, validation and testing sets were constituted randomly where minority and majority class samples were distributed equally.

The proposed method is compared with the state of the art methods given in Table 4 in combination with feature selection and with imbalanced data set handling approaches: SMOTE and Under Bagging (Balanced Training). For each method the same training, validation and testing sets were used. Therefore, for the standard version of the methods (kNN, C4.5, NB, SVM, RF BT and proposed) and all versions of them with SMOTE we used the same training and testing sets. On the other hand, for the experiments with feature selection we used validation sets as well to pick the best feature set for

each method on each data set (except the proposed method which uses the training set to pick the best feature subset).

Table 4: State-of-the-art methods and their combinations with preprocessing algorithms that are used.

Method	Parameters	Abbreviation
k-Nearest Neighbors	*k={1, 2, 5, 10, 15, 20, 25}* were used as the common parameters. For any dataset which performs best with *k*=25, larger *k* values until 50 were tested. For any *k* value which gave local maximum we applied intermediate k values as well. For instance, if we obtain the best performance when *k*=5 but the performance decreased sharply when *k*=10 then we tried *k={6, 8}* as well (which did not happen a lot).	kNN
k- Nearest Neighbors with Feature Selection		kNN wFS
k- Nearest Neighbors with SMOTE		kNN SMOTE
k- Nearest Neighbors with SMOTE and Feature Selection		kNN SMOTE wFS
C4.5	We used Quilan's C4.5 [66] code. Percentage of incorrectly assigned samples at a node (confidence level) was taken as *{0.05, 0.1, 0.2, 0.3}*.	C4.5
C4.5 with SMOTE		C4.5 SMOTE
Naive Bayes	As distributions: the normal distribution, kernel density estimation with different kernels such as normal, box, Epanechnikov etc. were tested with equal prior probabilities.	NB
Naive Bayes with Feature Selection		NB wFS
NB with SMOTE		NB SMOTE
NB with SMOTE and Feature Selection		NB SMOTE wFS
SVM	As the kernel function, a radial basis function with varying kernel parameters was used. Hyperplanes were separated by Sequential Minimal Optimization.	SVM
SVM with Feature Selection		SVM wFS
SVM with SMOTE		SVM SMOTE
SVM with SMOTE and Feature Selection		SV SMOTE wFS
Random Forest with Balanced Training	A number of trees *{10, 50, 100, 150, 200, 500, 1000, 2000}* were tested and the trees are grown without pruning. For node splitting, the Gini index [67] was used.	RF BT
Random Forest with Balanced Training and Feature Selection		RF BT wFS
Proposed Method	As the outlier detection parameter *{-1, -0.3, 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.9, 1, 1.5, 2, 2.5, 3}* were tested.	Proposed

The best results in terms of average geometric mean of sensitivity and specificity (GeoMean) [8] (Eq. 3), average adjusted geometric mean (AGeoMean) [9] (Eq. 4 where *Nn* refers to proportion of the majority samples) and Area Under Curve (AUC) [10] (Eq. 5) were given in Tables 5, 6, 7 and 8. For each evaluation metric the standard deviation (considering the folds in cross validation) is also given after the ± sign.

$$\textit{Geometric Mean of Sensitivity and Specificity (GeoMean)} = \sqrt{TPrate \; x \; TNrate} \tag{3}$$

$$\textit{(AGeoMean)} = \begin{cases} (GeoMean + TNrate \; x \; Nn)/(1 + Nn) & , TPrate > 0 \\ 0, & TPrate = 0 \end{cases} \tag{4}$$

Adjusted Geometric Mean

Nn=proportion of the negative class (majority) examples

$$\textit{AUC} = (1+TPrate-FPrate) / 2 \tag{5}$$

$$FPrate = FP / (FP+TN)$$

Table 5: Results (given the best parameter setting) of the methods which never presented the best performance in any of the dataset in terms of the average GeoMean (top of 3 values), AGeoMean (middle) and AUC (bottom) respectively.

Data Set	kNN	kNN wFS	NB	NB wFS	NB SMOTE wFS	Data Set	kNN	kNN wFS	NB	NB wFS	NB SMOTE wFS
	0.54±0.01	0.47±0.04	0.66±0.01	0.66±0.01	0.64±0.01		0.74±0.43	0.53±0.50	0.39±0.54	0.12±0.26	0.52±0.48
Vehicle1	0.68±0.01	0.66±0.01	0.68±0.01	0.63±0.05	0.65±0.02	**Glass5**	0.76±0.43	0.56±0.51	0.39±0.53	0.13±0.28	0.53±0.49
	0.54±0.04	0.56±0.03	0.66±0.03	0.67±0.01	0.64±0.02		0.50±0.01	0.48±0.03	0.69±0.27	0.48±0.07	0.70±0.25
	0.20±0.28	0.55±0.21	0.20±0.29	0.36±0.05	0.46±0.05		0.89±0.06	0.86±0.10	0.89±0.12	0.82±0.17	0.89±0.17
Yeast-1-2-8-9_vs_7	0.33±0.47	0.77±0.12	0.35±0.49	0.58±0.14	0.49±0.01	**Ecoli4**	0.94±0.03	0.93±0.05	0.93±0.06	0.86±0.13	0.91±0.14
	0.50±0.00	0.50±0.00	0.53±0.03	0.48±0.01	0.47±0.02		0.50±0.01	0.49±0.01	0.90±0.10	0.83±0.16	0.89±0.16
	0.10±0.22	0.00±0.00	0.11±0.24	0.59±0.10	0.45±0.28		0.78±0.12	0.74±0.08	0.84±0.08	0.73±0.11	0.82±0.05
Winequality-red-8_vs_6-7	0.15±0.33	0.00±0.00	0.14±0.32	0.60±0.10	0.60±0.10	**Ecoli-0-1-4-7_vs_2-3-5-6**	0.88±0.06	0.85±0.04	0.87±0.05	0.76±0.12	0.82±0.06
	0.50±0.00	0.00±0.00	0.48±0.07	0.61±0.07	0.52±0.17		0.49±0.01	0.49±0.01	0.85±0.08	0.74±0.11	0.83±0.05
	0.86±0.10	0.75±0.13	0.86±0.08	0.76±0.14	0.72±0.15		0.66±0.17	0.73±0.10	0.68±0.17	0.66±0.11	0.66±0.06
Ecoli-0-1-4-7_vs_5-6	0.92±0.05	0.85±0.08	0.90±0.05	0.77±0.12	0.78±0.09	**Appendicitis**	0.60±0.18	0.69±0.07	0.61±0.18	0.60±0.16	0.61±0.16
	0.51±0.05	0.49±0.01	0.87±0.08	0.77±0.14	0.74±0.13		0.91±0.06	0.92±0.08	0.70±0.16	0.67±0.12	0.69±0.11
	0.36±0.34	0.28±0.39	0.84±0.12	0.63±0.36	0.73±0.10		0.68±0.04	0.58±0.11	0.72±0.04	0.65±0.07	0.70±0.05
Cleveland-0_vs_4	0.46±0.42	0.33±0.46	0.88±0.06	0.65±0.37	0.76±0.06	**Hepato**	0.78±0.03	0.71±0.06	0.78±0.03	0.67±0.05	0.70±0.04
	0.50±0.01	0.50±0.01	0.85±0.14	0.72±0.17	0.74±0.10		0.35±0.05	0.29±0.07	0.76±0.04	0.66±0.06	0.70±0.05
	0.43±0.41	0.11±0.25	0.72±0.04	0.55±0.31	0.68±0.16		0.80±0.05	0.82±0.04	0.90±0.03	0.82±0.02	0.81±0.08
Glass2	0.50±0.46	0.15±0.33	0.65±0.04	0.52±0.29	0.66±0.14	**Ionosphere**	0.75±0.06	0.79±0.06	0.90±0.03	0.79±0.03	0.80±0.08
	0.50±0.01	0.50±0.00	0.73±0.08	0.70±0.15	0.70±0.15		0.88±0.02	0.90±0.08	0.90±0.03	0.83±0.01	0.80±0.08
	0.81±0.01	0.68±0.10	0.90±0.03	0.86±0.06	0.93±0.07		0.47±0.25	0.54±0.16	0.68±0.05	0.79±0.07	0.78±0.07
Yeast5	0.90±0.01	0.84±0.05	0.93±0.02	0.90±0.03	0.93±0.04	**Oil**	0.20±0.19	0.43±0.16	0.55±0.07	0.71±0.21	0.76±0.11
	0.72±0.01	0.55±0.01	0.90±0.04	0.83±0.04	0.90±0.07		0.58±0.01	0.58±0.01	0.69±0.03	0.76±0.13	0.79±0.06
	0.38±0.23	0.28±0.26	0.53±0.07	0.53±0.07	0.52±0.14		0.66±0.03	0.68±0.02	0.60±0.07	0.60±0.06	0.62±0.06
Yeast-1_vs_7	0.56±0.32	0.43±0.40	0.74±0.04	0.74±0.04	0.54±0.14	**Pima**	0.61±0.04	0.63±0.02	0.72±0.11	0.69±0.08	0.65±0.06
	0.50±0.01	0.50±0.01	0.63±0.04	0.63±0.04	0.55±0.12		0.71±0.05	0.70±0.07	0.64±0.06	0.68±0.09	0.64±0.06
	0.90±0.03	0.78±0.03	0.84±0.03	0.88±0.06	0.87±0.08		0.81±0.02	0.66±0.06	0.82±0.02	0.85±0.02	0.84±0.01
Vehicle2	0.92±0.02	0.85±0.06	0.86±0.03	0.86±0.06	0.85±0.08	**Satimage**	0.75±0.03	0.57±0.07	0.86±0.03	0.87±0.02	0.86±0.01
	0.76±0.04	0.75±0.05	0.84±0.03	0.89±0.06	0.87±0.08		0.90±0.004	0.85±0.007	0.85±0.02	0.85±0.02	0.84±0.01
	0.89±0.04	0.81±0.05	0.76±0.02	0.77±0.03	0.77±0.02		0.40±0.14	0.33±0.06	0.46±0.07	0.64±0.01	0.61±0.02
Vehicle0	0.91±0.02	0.85±0.03	0.68±0.03	0.71±0.04	0.70±0.02	**Scene**	0.30±0.12	0.23±0.05	0.35±0.07	0.60±0.04	0.62±0.02
	0.84±0.05	0.81±0.05	0.79±0.02	0.79±0.02	0.79±0.02		0.58±0.01	0.53±0.01	0.58±0.03	0.64±0.01	0.63±0.02

Table 6: Results (given the best parameter setting) of the other methods in terms of the average GeoMean. The best results on each data set are emphasized in bold-face.

Data Set	kNN SMOTE	kNN SMOTE wFS	C4.5	C4.5 SMOTE	NB SMOTE	SVM	SVM wFS	SVM SMOTE	SVM SMOTE wFS	RF BT	RF BT wFS	Proposed
Vehicle1	0.69±0.01	0.63±0.03	0.52±0.09	0.66±0.01	0.68±0.01	**0.77±0.01**	0.68±0.08	**0.77±0.01**	0.63±0.06	0.72±0.06	0.61±0.05	0.70±0.05
Yeast-1-2-8-9_vs_7	0.75±0.07	0.49±0.13	0.19±0.27	0.72±0.01	0.49±0.12	0.75±0.06	0.62±0.10	0.69±0.02	0.66±0.13	0.71±0.01	0.56±0.02	**0.82±0.06**
Winequality-red-8_vs_6-7	0.23±0.31	0.35±0.34	0.10±0.22	0.41±0.23	0.56±0.15	0.62±0.20	0.71±0.11	0.76±0.08	0.65±0.12	0.77±0.05	0.60±0.07	**0.81±0.10**
Ecoli-0-1-4-7_vs_5-6	0.89±0.05	0.79±0.10	0.85±0.12	0.82±0.06	0.83±0.11	0.87±0.08	0.79±0.09	0.80±0.11	0.72±0.10	0.87±0.11	0.76±0.14	**0.94±0.06**
Cleveland-0_vs_4	0.57±0.35	0.74±0.17	0.32±0.44	0.32±0.07	0.78±0.17	**0.90±0.18**	0.77±0.12	0.85±0.05	0.81±0.08	0.87±0.18	0.62±0.35	0.85±0.08
Glass2	0.79±0.10	0.65±0.13	0.00±0.00	0.00±0.00	0.72±0.07	0.76±0.15	0.64±0.09	0.74±0.13	0.65±0.05	0.76±0.08	0.66±0.18	**0.81±0.10**
Yeast5	0.96±0.01	**0.97±0.01**	0.52±0.17	0.95±0.01	0.80±0.07	**0.97±0.01**	0.96±0.01	0.87±0.01	0.87±0.02	**0.97±0.01**	**0.97±0.01**	0.93±0.10
Yeast-1_vs_7	0.69±0.09	0.64±0.09	0.31±0.29	0.54±0.05	0.57±0.15	0.76±0.04	0.70±0.10	0.68±0.04	0.62±0.05	0.76±0.06	0.65±0.05	**0.80±0.16**
Vehicle2	0.91±0.03	0.84±0.09	0.64±0.03	0.81±0.03	0.85±0.03	**0.97±0.02**	0.90±0.03	0.96±0.03	0.88±0.04	**0.97±0.01**	0.91±0.04	**0.97±0.05**
Vehicle0	0.92±0.02	0.88±0.02	0.89±0.02	0.90±0.03	0.78±0.02	**0.96±0.01**	0.84±0.05	0.93±0.03	0.80±0.07	**0.96±0.01**	0.89±0.03	**0.96±0.05**
Glass5	0.83±0.16	0.87±0.16	0.00±0.00	0.00±0.00	0.39±0.54	0.92±0.13	0.90±0.05	0.88±0.13	**0.95±0.04**	0.89±0.06	0.92±0.06	0.88±0.13
Ecoli4	0.97±0.02	0.91±0.10	0.85±0.09	0.91±0.07	0.88±0.11	0.97±0.06	0.92±0.08	0.90±0.07	0.83±0.10	0.92±0.07	0.83±0.19	**0.99±0.02**
Ecoli-0-1-4-7_vs_2-3-5-6	0.89±0.05	0.80±0.08	0.80±0.15	0.84±0.07	0.78±0.14	0.91±0.08	0.77±0.08	0.78±0.08	0.72±0.08	0.87±0.09	0.79±0.09	**0.92±0.08**
Appendicitis	0.76±0.03	0.75±0.04	0.69±0.01	0.73±0.07	0.68±0.16	0.74±0.08	0.79±0.01	0.74±0.08	0.78±0.01	0.72±0.05	0.77±0.04	**0.83±0.07**
Hepato	**0.73±0.06**	0.65±0.07	0.67±0.03	**0.73±0.05**	**0.73±0.06**	**0.73±0.06**	0.70±0.05	0.45±0.01	0.69±0.07	0.73±0.03	0.69±0.07	**0.73±0.04**
Ionosphere	0.86±0.05	0.90±0.06	0.56±0.34	0.80±0.09	0.88±0.02	0.88±0.04	0.89±0.01	**0.92±0.03**	0.91±0.02	0.91±0.04	0.89±0.05	0.87±0.05
Oil	0.76±0.04	0.77±0.03	0.60±0.15	0.77±0.05	0.65±0.08	0.78±0.07	0.78±0.04	0.41±0.19	0.77±0.13	0.77±0.05	0.79±0.05	**0.82±0.08**
Pima	0.72±0.02	0.72±0.04	0.62±0.05	0.70±0.04	0.72±0.07	0.73±0.03	0.71±0.02	0.74±0.02	0.73±0.03	0.73±0.04	0.68±0.08	**0.79±0.05**
Satimage	0.89±0.02	0.76±0.02	0.61±0.07	0.82±0.01	0.85±0.02	**0.90±0.02**	0.86±0.01	0.56±0.03	0.83±0.02	0.88±0.02	0.85±0.02	0.84±0.10
Scene	0.66±0.04	0.53±0.04	0.42±0.01	0.67±0.03	0.55±0.08	0.53±0.13	0.67±0.06	0.68±0.05	0.70±0.10	**0.72±0.02**	0.64±0.05	0.61±0.13

Table 7: Results (given the best parameter setting) of the other methods in terms of the average AGeoMean. The best results on each data set are emphasized in bold-face.

Data Set	kNN SMOTE	kNN SMOTE wFS	C4.5	C4.5 SMOTE	NB SMOTE	SVM	SVM wFS	SVM SMOTE	SVM SMOTE wFS	RF BT	RF BT wFS	Proposed
Vehicle1	0.68±0.01	0.64±0.01	**0.81±0.03**	0.65±0.01	0.70±0.01	0.76±0.01	0.69±0.05	0.75±0.01	0.68±0.05	0.70±0.01	0.61±0.01	0.71±0.08
Yeast-1-2-8-9_vs_7	0.75±0.04	0.78±0.08	0.32±0.45	0.77±0.05	0.72±0.07	**0.79±0.02**	0.69±0.04	0.64±0.05	0.57±0.05	0.74±0.01	0.59±0.04	0.75±0.05
Winequality-red-8_vs_6-7	0.29±0.39	0.44±0.41	0.15±0.33	0.56±0.32	0.14±0.32	0.70±0.08	0.43±0.27	**0.72±0.11**	0.69±0.12	0.71±0.04	0.57±0.09	0.70±0.10
Ecoli-0-1-4-7_vs_5-6	0.92±0.02	0.82±0.08	0.90±0.06	0.88±0.04	0.90±0.06	0.90±0.04	0.74±0.13	0.79±0.06	0.66±0.09	0.87±0.02	0.79±0.13	**0.96±.0.04**
Cleveland-0_vs_4	0.62±0.35	0.80±0.12	0.35±0.48	0.25±0.04	0.84±0.09	**0.93±0.10**	0.76±0.09	0.85±0.09	0.79±0.05	0.89±0.09	0.63±0.36	**0.93±0.08**
Glass2	0.80±0.04	0.67±0.09	0.00±0.00	0.00±0.00	0.71±0.05	**0.84±0.07**	0.64±0.11	0.77±0.07	0.61±0.12	0.71±0.07	0.62±0.14	0.83±0.09
Yeast5	0.95±0.01	0.95±0.01	0.76±0.09	0.93±0.01	0.88±0.03	0.95±0.01	0.94±0.01	0.82±0.01	0.81±0.03	**0.96±0.01**	0.95±0.01	0.92±0.10
Yeast-1_vs_7	0.72±0.04	0.70±0.08	0.44±0.40	0.60±0.06	0.74±0.09	0.77±0.03	0.73±0.04	0.62±0.07	0.59±0.10	0.76±0.04	0.62±0.07	**0.88±0.10**
Vehicle2	0.92±0.02	0.85±0.08	0.79±0.01	0.78±0.03	0.88±0.03	**0.97±0.01**	0.88±0.04	0.96±0.02	0.86±0.04	**0.97±0.02**	0.90±0.04	0.93±0.07
Vehicle0	0.92±0.02	0.86±0.04	0.87±0.02	0.90±0.02	0.70±0.03	**0.97±0.01**	0.88±0.04	0.96±0.02	0.86±0.04	0.95±0.01	0.87±0.04	**0.97±0.02**
Glass5	0.85±0.12	0.92±0.09	0.00±0.00	0.54±0.49	0.39±0.53	**0.94±0.08**	0.90±0.16	0.93±0.08	0.92±0.06	0.84±0.08	0.89±0.08	0.83±0.06
Ecoli4	0.96±0.02	0.92±0.07	0.91±0.04	0.95±0.04	0.93±0.05	0.97±0.03	0.94±0.05	0.88±0.06	0.78±0.09	0.94±0.04	0.89±0.09	**0.99±0.01**
Ecoli-0-1-4-7_vs_2-3-5-6	**0.92±0.03**	0.84±0.03	0.89±0.07	0.89±0.04	0.78±0.17	**0.92±0.05**	0.82±0.06	0.82±0.05	0.70±0.02	0.87±0.08	0.82±0.06	0.90±0.09
Appendicitis	0.75±0.03	0.74±0.02	0.77±0.06	0.78±0.04	0.62±0.17	0.70±0.07	0.77±0.04	0.69±0.07	0.78±0.08	0.68±0.05	0.75±0.02	**0.91±0.08**
Hepato	0.77±0.04	0.70±0.04	0.61±0.04	0.71±0.10	0.75±0.04	0.74±0.06	0.71±0.08	0.34±0.01	0.70±0.06	0.81±0.02	0.67±0.08	**0.92±0.07**
Ionosphere	0.82±0.06	0.88±0.07	0.64±0.37	0.84±0.07	0.90±0.02	0.95±0.02	0.88±0.06	0.95±0.03	0.89±0.02	0.91±0.04	0.91±0.04	**0.99±0.01**
Oil	0.72±0.06	0.81±0.06	0.79±0.08	0.82±0.04	0.55±0.09	0.72±0.09	0.83±0.09	0.11±0.24	0.71±0.15	0.83±0.08	0.86±0.07	**0.99±0.01**
Pima	0.76±0.04	0.74±0.04	0.73±0.03	0.70±0.03	0.75±0.08	0.77±0.06	0.73±0.02	0.77±0.05	0.75±0.03	0.77±0.05	0.69±0.09	**0.88±0.11**
Satimage	0.89±0.03	0.71±0.03	0.76±0.04	0.84±0.01	0.85±0.03	**0.90±0.02**	**0.90±0.02**	**0.90±0.02**	0.87±0.02	0.89±0.03	0.86±0.03	**0.90±0.10**
Scene	0.63±0.06	0.49±0.05	0.67±0.06	0.70±0.03	0.46±0.10	0.55±0.13	0.71±0.06	0.60±0.05	0.74±0.10	0.72±0.05	0.65±0.08	**0.83±0.01**

Table 8: Results (given the best parameter setting) of the other methods in terms of the average AUC. The best results on each data set are emphasized in bold-face.

Data Set	kNN SMOTE	kNN SMOTE wFS	C4.5	C4.5 SMOTE	NB SMOTE	SVM	SVM wFS	SVM SMOTE	SVM SMOTE wFS	RF BT	RF BT wFS	Proposed
Vehicle1	0.65±0.02	0.54±0.04	0.56±0.07	0.61±0.04	0.68±0.03	0.66±0.16	0.65±0.04	0.76±0.12	0.62±0.03	0.72±0.05	0.64±0.04	0.69±0.02
Yeast-1-2-8-9_vs_7	0.73±0.09	0.64±0.07	0.50±0.00	0.51±0.00	0.61±0.03	0.80±0.11	0.67±0.05	0.82±0.15	0.57±0.08	0.71±0.04	0.53±0.12	**0.86±0.03**
Winequality-red-8_vs_6-7	0.51±0.06	0.54±0.13	0.50±0.00	0.55±0.12	0.62±0.11	0.64±0.21	0.62±0.10	0.72±0.19	0.71±0.16	**0.75±0.08**	0.63±0.12	0.74±0.10
Ecoli-0-1-4-7_vs_5-6	0.92±0.06	0.82±0.06	0.60±0.17	0.77±0.11	0.84±0.10	0.88±0.08	0.63±0.29	0.92±0.11	0.72±0.11	0.86±0.10	0.74±0.07	**0.96±0.13**
Cleveland-0_vs_4	0.64±0.19	0.60±0.11	0.55±0.14	0.37±0.13	0.80±0.18	**0.91±0.14**	0.70±0.21	0.89±0.17	0.53±0.35	0.84±0.12	0.62±0.17	**0.91±0.09**
Glass2	0.82±0.07	0.59±0.08	0.09±0.02	0.09±0.02	0.73±0.08	0.56±0.34	0.71±0.22	0.70±0.28	0.63±0.27	0.77±0.08	0.62±0.16	**0.83±0.05**
Yeast5	0.98±0.01	0.95±0.04	0.50±0.00	0.97±0.01	0.79±0.03	0.96±0.02	**0.99±0.01**	0.91±0.12	0.82±0.26	0.97±0.01	0.95±0.03	0.95±0.04
Yeast-1_vs_7	0.70±0.08	0.65±0.09	0.50±0.00	0.57±0.07	0.65±0.10	0.77±0.07	0.67±0.11	0.72±0.18	0.59±0.10	0.73±0.03	0.65±0.07	**0.81±0.03**
Vehicle2	0.86±0.02	0.83±0.07	0.67±0.04	0.81±0.03	0.85±0.03	**0.97±0.01**	0.83±0.06	0.96±0.02	0.86±0.06	**0.97±0.02**	0.91±0.04	**0.97±0.14**
Vehicle0	0.95±0.005	0.89±0.02	0.86±0.07	0.82±0.07	0.80±0.01	0.95±0.01	0.83±0.04	0.90±0.06	0.82±0.06	**0.97±0.004**	0.90±0.01	0.96±0.02
Glass5	0.88±0.17	0.80±0.21	0.09±0.02	0.09±0.02	0.69±0.27	**0.94±0.09**	0.82±0.27	0.89±0.15	0.91±0.16	0.89±0.05	0.89±0.05	0.89±0.16
Ecoli4	**0.99±0.01**	0.92±0.06	0.58±0.16	0.85±0.05	0.89±0.10	0.97±0.08	0.83±0.26	0.92±0.09	0.91±0.06	0.93±0.07	0.85±0.14	**0.99±0.10**
Ecoli-0-1-4-7_vs_2-3-5-6	0.89±0.05	0.82±0.03	0.58±0.14	0.78±0.11	0.79±0.12	0.90±0.09	0.64±0.11	0.78±0.15	0.73±0.10	0.87±0.08	0.80±0.10	0.93±0.04
Appendicitis	0.79±0.12	0.78±0.08	0.55±0.07	0.59±0.01	0.71±0.14	0.89±0.09	0.75±0.16	0.83±0.14	0.84±0.09	0.79±0.09	0.84±0.16	0.96±0.05
Hepato	0.66±0.07	0.55±0.10	0.76±0.02	**0.77±0.06**	0.74±0.05	0.67±0.07	0.54±0.12	0.20±0.27	0.66±0.08	0.77±0.04	0.72±0.07	0.76±0.03
Ionosphere	**0.96±0.03**	0.84±0.08	0.67±0.12	0.70±0.16	0.89±0.02	0.89±0.05	**0.96±0.03**	0.87±0.05	0.91±0.06	0.89±0.05	0.84±0.04	**0.96±0.02**
Oil	0.82±0.02	0.81±0.04	0.64±0.08	0.77±0.11	0.69±0.05	**0.97±0.01**	0.86±0.04	0.50±0.00	0.79±0.05	0.84±0.04	0.82±0.08	**0.97±0.01**
Pima	0.72±0.03	0.67±0.01	0.63±0.03	0.70±0.05	0.73±0.07	0.74±0.05	0.71±0.08	0.77±0.04	0.75±0.06	0.72±0.04	0.63±0.04	**0.79±0.06**
Satimage	**0.91±0.01**	0.88±0.01	0.80±0.05	0.88±0.05	0.85±0.02	0.90±0.01	0.80±0.02	0.50±0.01	0.82±0.01	0.87±0.01	0.81±0.02	0.85±0.02
Scene	0.73±0.02	0.63±0.06	0.49±0.05	0.61±0.09	0.61±0.05	0.60±0.01	0.62±0.06	**0.76±0.01**	0.62±0.06	0.73±0.02	0.63±0.03	0.64±0.002

Table 5 shows the results (with the best parameter setting) of the methods which never produce the best result in any of the datasets. Results are given in terms of the average GeoMean, AGeoMean and AUC respectively. Tables 6, 7 and 8 gives the results (for the best parameter setting) in terms of GeoMean, AGeoMean and AUC for algorithms that performed best for at least one dataset in terms of one metric. In these tables, the best results on each data set are emphasized in bold-face

The results show that the performance of the proposed method was the best on 13 of 20 datasets for GeoMean, 12 of 20 datasets for AGeoMean and 10 of 20 datasets for AUC out of 16 other classification methods. The next best method was SVM with (7, 8, 4) out of 20 datasets in terms of GeoMean, AGeoMean and AUC respectively. All other methods were worse. The proposed method generally performs better in terms of GeoMean, if the IR is low (such as Winequality-red-8_vs_6-7, Yeast-1-2-8-9_vs_7 and Oil). The high performance in terms of AGeoMean also shows that it is good at majority class classification while as good as other methods for classification of minority class (can be infer from GeoMean results). Additionally, the proposed method performs well enough in terms of AUC which can be supported by the statistical tests results given in the next section. Average results over the 20 datasets also show that the proposed method is the best method for each of the three metrics.

4.1.3 Statistical Tests

To compare the different methods appropriately, we applied a couple of statistical tests to the GeoMean, AGeoMean, and AUC results. We carried out parametric and non-parametric tests as suggested in the literature [68, 69, 70] and applied in other papers related to imbalanced data set classification such as [2]. We used paired *t*-test as pair wise comparison test to find out if there is a significant difference between a pair of methods. As a multiple comparison test, we applied the Friedman test [70] to determine the statistical significance between methods given in Table 4. When we found statistical difference between the methods and the proposed method, we applied the Holm post hoc test [71] to test if the proposed method is significantly better than the others or not.

- *Paired t-test [70]:* It considers the differences between the paired values in two data sets by looking at the variation of corresponding values and produces a value (*p*-value) which determines how likely it is that the two values are from the same population [68, 70]. The paired values are the performances on each fold from the two compared algorithm. This test can be used to justify if the performances of two algorithms are significantly different or not. The *p*-value which determines if the comparison is significant or not and also indicates how significant it is (If the proposed method is better, then the smaller the *p*-value, the more significantly better it is. Conversely, if any other method is better, then the smaller value of *p* shows how much better it is than the proposed method). For all tests, the significance level is taken as 0.05.

- *Friedman test with Iman-Davenport Extension [70]:* Methods are ranked on each data set according to their performance (best performance takes the lowest rank). For each classifier the sum of its ranks on all data set is calculated. Following this, Friedman's and Iman-Davenport statistics are calculated using the formulas given in [70]. The results of statistics are compared with the corresponding value in the F-distribution table and if the value in the table is smaller than the found statistics, this means that the null hypothesis (all classifiers perform

the same and the observed differences are random) of Friedman is rejected and there is a significant difference between the algorithms.

- *Holm post hoc test [71]:* Is based on the value *z* given in Eq. 6, p-value obtained from normal distribution corresponds to z and the adjusted alpha (described below). A p-value smaller than the corresponding adjusted alpha means that the null hypothesis is rejected. Hence, there is a significant difference between the compared methods. Otherwise, the null hypothesis is not rejected which means that we can stop checking the hypothesis since the hypothesis for further methods (better in performance, which has p-value) should already be as not rejected.

The average ranks used in the computation of the Friedman test for the metrics GeoMean, AGeoMean and AUC are shown in Table 9. In this table, the best rank (smallest one) is shown in bold. The Friedman and Iman Davenport statistics are also given for the performance results in terms of GeoMean, AGeoMean and AUC. For the calculation using GeoMean, the critical value of F distribution (16,304) is 2.01 when α=0.05, which is smaller than the Iman-Davenport (18.47) statistic meaning that the null hypothesis of Friedman (given above) is rejected by a high level of significance. Similarly, for AGeoMean and AUC, Iman-Davenport statistic (10.81, and 16.20 respectively) are larger than 2.01 which also rejects the null hypothesis. Since the Friedman test results showed a high significance, we applied the post hoc Holm test.

Table 9: The average ranks used in the computation of the Friedman test for the metrics GeoMean, AGeoMean and AUC respectively. Lower rank means better performance. The best performance is shown in bold.

AVERAGE RANK USING ALL DATASETS	***kNN***	***kNN wFS***	***kNN SMOTE***	***kNN SMOTE wFS***	***C4.5***	***C4.5 SMOTE***	***NB***	***NB wFS***	***NB SMOTE***
GeoMean	12.85	14.58	5.95	9.63	14.75	10.18	10.6	11.83	9.93
AGeoMean	11.43	12.98	6.53	9.35	11.75	9.48	10.23	12.58	9.9
AUC	13.05	13.8	5.45	9.48	14.65	11.42	9.6	10.82	8.85
	NB SMOTE wFS	***SVM***	***SVM wFS***	***SVM SMOTE***	***SVM SMOTE wFS***	***RF BT***	***RF BT wFS***	***Proposed***	
GeoMean	11.1	3.75	6.83	7.28	8.55	3.75	8.08	**3.4**	
AGeoMean	11.88	3.65	7.90	7.98	10.18	4.83	9.63	**2.78**	
AUC	10.9	4.6	8.95	6.73	9.18	4.05	8.9	**2.58**	
Friedman statistic for **GeoMean**= 157.72 Iman-Davenport statistic for **GeoMean**=18.47 F (16,304) =2.01, α=0.05, **the null hypothesis of Friedman is rejected by a high level of significance.**									
Friedman statistic for **AGeoMean**= 116.01 Iman-Davenport statistic for **AGeoMean**= 10.81 F (16,304) =2.01, α=0.05, **the null hypothesis of Friedman is rejected by a high level of significance.**									
Friedman statistic for **AUC**= 147.26 Iman-Davenport statistic for **AUC**=16.20 F (16,304) =2.01, α=0.05, **the null hypothesis of Friedman is rejected by a high level of significance**.									

Tables 10, 11 and 12 show the Holm test results using the GeoMean, AGeoMean and AUC performance results respectively. Methods kNN, kNN wFS, NB, NB wFS, NB SMOTE wFS were not

compared as they did not perform well enough to justify the statistical analysis and they also perform significantly worse than the proposed algorithm on all evaluation metrics. Comparing fewer methods is better as the Holm test is affected by the number of methods compared (comparing more methods might show the proposed method is more successful than it is). In these tables, z_i is calculated as given in Eq. 6 where k refers to the number of methods (which is 11), N refers to the number of datasets (which is 20), $R_{proposed}$ refers to ranking of the proposed method in terms of different evaluation metrics when 11 methods are used (should be taken as the given value in Table 9 if 16 methods are used for Holm test) and R_i is the ranking of the i^{th} method in terms of different evaluation metrics out of 11 methods (should be taken as the value given in Table 9 if 16 methods are used for Holm test). The p-value is based on the normal distribution and Holm adjusted alpha (shown as Holm) equals to $0.005/i$. Hypothesis given as rejected means a significant difference between the compared methods and this happens if the p-value is smaller than the corresponding Holm value. Negative values of z means that the proposed method performed better than the compared method.

$$z_i = (R_{proposed} - R_i) / \sqrt{\frac{k\,(k+1)}{6N}} \tag{6}$$

Table 10: Holm test results for the comparison between proposed method and the other methods using GeoMean.

i	Methods	z_i	p_value	Holm	Hypothesis
11	C4.5	-7.4132	0.0001	0.0045	**Rejected for Proposed**
10	C4.5 SMOTE	-4.9580	0.0001	0.0050	**Rejected for Proposed**
9	NB SMOTE	-4.9342	0.0001	0.0056	**Rejected for Proposed**
8	kNN SMOTE wFS	-4.6720	0.0001	0.0063	**Rejected for Proposed**
7	SVM SMOTE wFS	-3.6708	0.0002	0.0071	**Rejected for Proposed**
6	RF BT wFS	-3.6470	0.0003	0.0083	**Rejected for Proposed**
5	SVM SMOTE	-2.8842	0.0039	0.0100	**Rejected for Proposed**
4	SVM wFS	-2.8604	0.0042	0.0125	**Rejected for Proposed**
3	kNN SMOTE	-2.2645	0.0235	0.0167	Not Rejected
2	RF BT	-0.4052	0.6853	0.0250	Not Rejected
1	SVM	-0.3337	0.7386	0.0500	Not Rejected

Table 11: Holm test results for the comparison between proposed method and the other methods using AGeoMean.

i	Methods	z_i	p_value	Holm	Hypothesis
11	C4.5	-6.1022	0.0001	0.0045	**Rejected for Proposed**
10	SVM SMOTE wFS	-5.4109	0.0001	0.0050	**Rejected for Proposed**
9	NB SMOTE	-5.1725	0.0001	0.0056	**Rejected for Proposed**
8	RF BT wFS	-4.9818	0.0001	0.0063	**Rejected for Proposed**
7	kNN SMOTE wFS	-4.8150	0.0001	0.0071	**Rejected for Proposed**
6	C4.5 SMOTE	-4.7911	0.0001	0.0083	**Rejected for Proposed**
5	SVM wFS	-3.9330	0.0001	0.0100	**Rejected for Proposed**
4	SVM SMOTE	-3.5993	0.0003	0.0125	**Rejected for Proposed**
3	kNN SMOTE	-3.0511	0.0023	0.0167	**Rejected for Proposed**
2	RF BT	-1.8116	0.0700	0.0250	Not Rejected
1	SVM	-0.6674	0.5045	0.0500	Not Rejected

Table 12: Holm test results for the comparison between proposed method and the other methods using AUC.

i	Methods	z_i	p_value	Holm	Hypothesis
11	C4.5	-8.2236	0.0001	0.0045	**Rejected for Proposed**
10	C4.5 SMOTE	-6.3644	0.0001	0.0050	**Rejected for Proposed**
9	kNN SMOTE wFS	-5.2679	0.0001	0.0056	**Rejected for Proposed**
8	NB SMOTE	-4.8865	0.0001	0.0063	**Rejected for Proposed**
7	SVM SMOTE wFS	-4.7911	0.0001	0.0071	**Rejected for Proposed**
6	SVM wFS	-4.7673	0.0001	0.0083	**Rejected for Proposed**
5	RF BT wFS	-4.6243	0.0001	0.0100	**Rejected for Proposed**
4	SVM SMOTE	-2.9796	0.0029	0.0125	**Rejected for Proposed**
3	kNN SMOTE	-2.2645	0.0235	0.0167	Not Rejected
2	SVM	-1.5017	0.1332	0.0250	Not Rejected
1	RF BT	-1.2395	0.2152	0.0500	Not Rejected

Tables 10, 11 and 12 show that the Holm test finds that the proposed method is the best over all comparisons to the best 11 other algorithms (all z values are negative) and it is significantly better than all methods except kNN SMOTE, RF BT and SVM in terms of GeoMean, and AUC and it is significantly better than all methods except RF BT and SVM in terms of AGeoMean. Although not shown, the Holm test applied to all 16 methods given in Table 4 show that the proposed method is significantly better than all methods except SVM SMOTE, SVM wFS, kNN SMOTE, RF BT and SVM in terms of GeoMean, is significantly better than all methods except RF BT, and SVM in terms of AGeoMean, and is significantly better than all methods except kNN SMOTE, SVM and RF BT in terms of AUC.

As a further statistical analysis, we applied paired *t*-test to see how well the proposed method performs compared to each other method for each dataset considering the performances in each cross validation fold. We used the results of GeoMean as it had the worst statistics for the proposed method when the

Holm test was applied. The paired *t*-test results between each method and the proposed method for GeoMean is given in Table 13 in terms of p-value. A p-value equal or smaller than 0.05 means there is

a significant difference. In this table, the results showing a significant advantage to the proposed method are shown in bold-face. Similarly, the results showing significantly worse performance by the proposed method are shown in italics (but there are no instances at this). High values of p (>0.5) mean that the two methods perform nearly the same. Mid values of p ($0.05<p<=0.5$) mean that the proposed method performs better for each fold, but the performance of the other method is also very close to the proposed method for at least one fold.

As seen in Table 13, the proposed method performed significantly better than the rest of the method in 94 tests out of 320 tests when each data set and pairs of methods are considered separately. On the other hand, it performed worse than another algorithm in 36 tests (out of 320 tests) and none of the methods performed significantly better than it (using GeoMean results). The proposed method never had significantly better performance than SVM and RF BT. It performed significantly better than kNN SMOTE, kNN SMOTE wFS, NB SMOTE wFS and SVM wFS only 2, 3, 4 and 3 times (out of 20) respectively.

In summary, in this section we compared the performance of the proposed method with the state of arts methods using 20 different public imbalanced dataset. The evaluation was done in terms of GeoMean, AGeoMean and AUC. In overall, the proposed method performed the best in all metrics. Statistical tests were also applied to the results. The Friedman test with Iman-Davenport Extension showed that the proposed method is significantly better than the rest using all metrics. However, the Holm test showed that the proposed method is not significantly better than all methods. The Holm test results were also confirmed by the paired t-test which is applied to GeoMean data. Paired t-test showed that the proposed method is not significantly better than SVM and RF BT while better on average in majority of the results.

Table 13: Paired *t*-test results between each method and the proposed method for GeoMean results.

Data Set	kNN	kNN wFS	kNN SMOTE	kNN SMOTE wFS	C4.5	C4.5 SMOTE	NB	NB wFS	NB SMOTE	NB SMOTE wFS	SVM	SVM wFS	SVM SMOTE	SVM SMOTE wFS	RF BT	RF BT wFS
Vehicle1	0.10	**0.02**	0.87	0.15	0.33	0.38	0.52	0.48	0.67	0.27	0.37	0.46	0.27	**0.04**	0.12	**0.02**
Yeast-1-2-8-9_vs_7	0.24	0.24	0.58	0.10	0.23	0.23	0.24	0.11	0.24	0.14	0.56	0.33	0.15	0.46	0.20	0.14
Winequality-red-8_vs_6-7	**0.01**	**0.00**	**0.03**	0.06	**0.01**	**0.03**	**0.00**	**0.01**	**0.01**	0.07	0.18	0.15	0.56	0.11	0.55	**0.00**
Ecoli-0-1-4-7_vs_5-6	0.17	**0.05**	0.20	**0.03**	0.14	**0.00**	0.06	0.09	0.11	0.06	0.18	**0.02**	**0.01**	**0.01**	0.14	**0.04**
Cleveland-0_vs_4	**0.03**	**0.02**	0.19	0.35	**0.03**	**0.00**	0.79	0.27	0.46	0.14	0.68	0.41	0.99	0.61	0.86	0.25
Glass2	0.10	**0.00**	0.09	0.11	**0.00**	**0.00**	0.11	0.14	0.07	0.22	0.36	0.06	0.11	**0.05**	0.26	0.19
Yeast5	0.17	**0.00**	0.53	0.50	0.10	0.61	0.61	0.33	**0.02**	0.97	0.51	0.59	0.37	0.48	0.47	0.50
Yeast-1_vs_7	**0.03**	**0.04**	0.08	0.20	**0.05**	**0.03**	**0.05**	**0.05**	0.13	**0.05**	0.54	0.38	0.13	**0.04**	0.63	**0.05**
Vehicle2	0.08	**0.00**	0.13	**0.04**	**0.00**	**0.00**	**0.01**	0.13	**0.02**	0.12	0.90	0.08	0.93	0.06	0.71	0.19
Vehicle0	**0.01**	**0.01**	0.08	0.06	**0.04**	0.06	**0.00**	**0.00**	**0.00**	**0.00**	0.91	**0.03**	0.47	**0.03**	0.97	0.08
Glass5	0.56	0.18	0.68	0.91	**0.00**	**0.00**	0.13	**0.00**	0.13	0.14	0.70	0.79	0.97	0.35	0.93	0.58
Ecoli4	**0.04**	0.06	0.33	0.16	**0.03**	0.09	0.15	0.10	0.11	0.26	0.50	0.13	0.08	**0.02**	0.09	0.15
Ecoli-0-1-4-7_vs_2-3-5-6	0.14	**0.04**	0.28	**0.03**	0.19	**0.00**	**0.00**	**0.04**	**0.03**	**0.04**	0.81	**0.00**	**0.04**	**0.02**	0.23	**0.06**
Appendicitis	0.26	0.13	0.25	0.12	0.19	**0.00**	0.29	0.19	0.27	0.17	0.08	0.56	0.08	0.55	0.06	0.19
Hepato	0.19	0.06	0.95	0.08	**0.01**	0.86	0.42	**0.03**	0.98	0.11	0.96	0.37	**0.00**	0.24	0.85	0.43
Ionosphere	0.08	0.19	0.67	0.58	0.13	0.15	0.30	0.04	0.68	0.24	0.86	0.43	0.17	0.36	0.30	0.78
Oil	**0.04**	**0.01**	0.17	0.30	**0.05**	0.34	**0.03**	0.19	**0.04**	0.24	0.39	0.39	**0.01**	0.44	0.22	0.50
Pima	**0.00**	**0.01**	**0.05**	0.06	**0.01**	**0.00**	**0.00**	**0.02**	0.13	**0.03**	0.08	0.09	**0.05**	0.08	0.09	**0.02**
Satimage	0.58	**0.03**	0.32	0.12	**0.02**	0.59	0.68	0.90	0.84	0.97	0.28	0.65	**0.01**	0.84	0.44	0.92
Scene	0.13	**0.03**	0.42	0.17	0.09	0.38	0.16	0.59	0.56	0.98	0.25	0.45	0.42	0.39	0.18	0.77

4.2 Experiments and Results with Synthetic Data Sets

To show the proposed method's performance in detail and understand in which specific conditions it performs better than the other methods we also created synthetic imbalanced data sets. The experiments were applied with data sets generated using Gaussian Mixture Models (GMM) and Uniform Mixture Models (UMM) with;

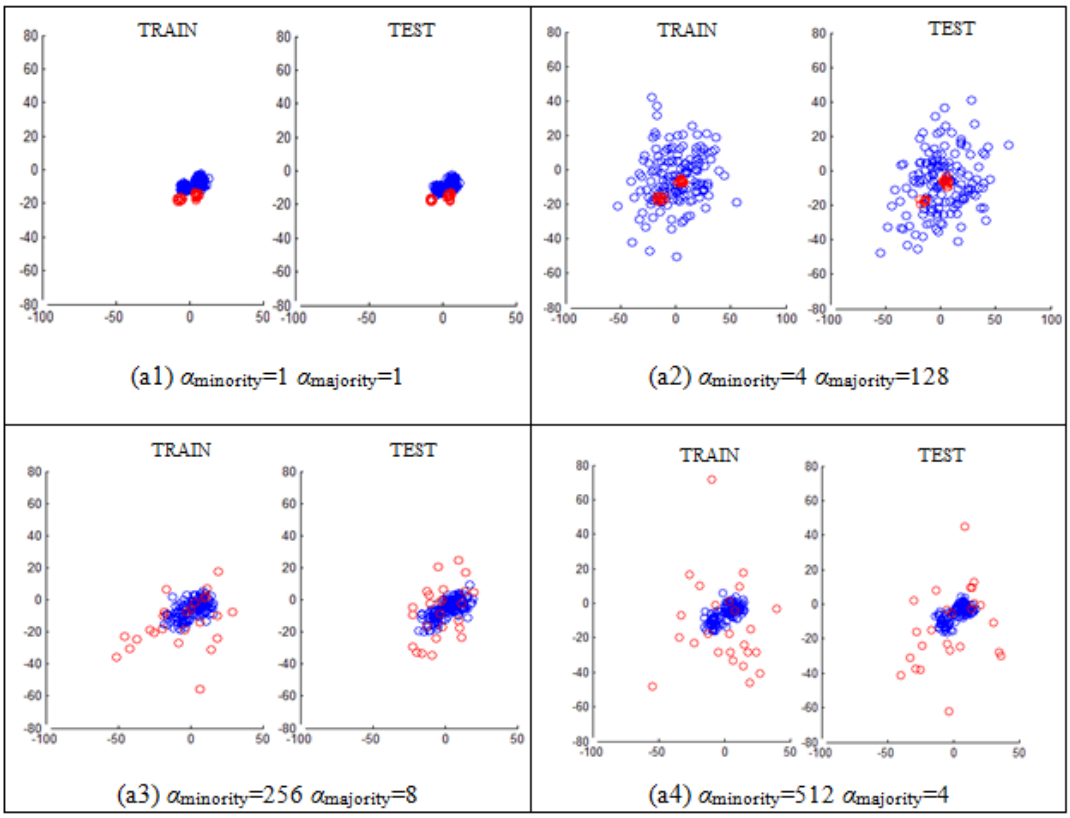
- Different number of features (2, 5 and 10),
- Different imbalanced ratios (0.67 (300 samples from majority class, 200 samples from minority class), 0.33 (300 samples from majority class, 100 from minority class) and 0.17 (300 samples from majority class, 50 samples from minority class))
- Different combinations for the standard deviation of majority and minority class distributions.

For both the majority and minority classes, a mixture of two equally weighted distributions was created as the baseline data set (see Figure 5a1; GMM baseline data set for 2 features, Figure 5b1; UMM baseline data set for 2 features). For experiments with the GMM; to create other data sets, we changed the co-variance of the components for each class by multiplying the variance of each component with a constant α coefficient while keeping the mean of each component constant. Then, we sampled the same number of samples with the baseline data set for both the majority class and the minority class. For small values of α, the majority and minority classes are tighter and separable as two different classes whereas for the bigger values of α, the data sets overlap (both the different components of classes and the classes themselves) and are sparser. In total, we obtained 100 different data sets while taking all pairs of $\alpha = \{1, 2, 4, 8, 16, 32, 64, 128, 256, 512\}$ for minority and majority classes. For the baseline data set, α_{minority} and α_{majority} are equal to 1 and mean and co-variance of distributions are selected randomly.

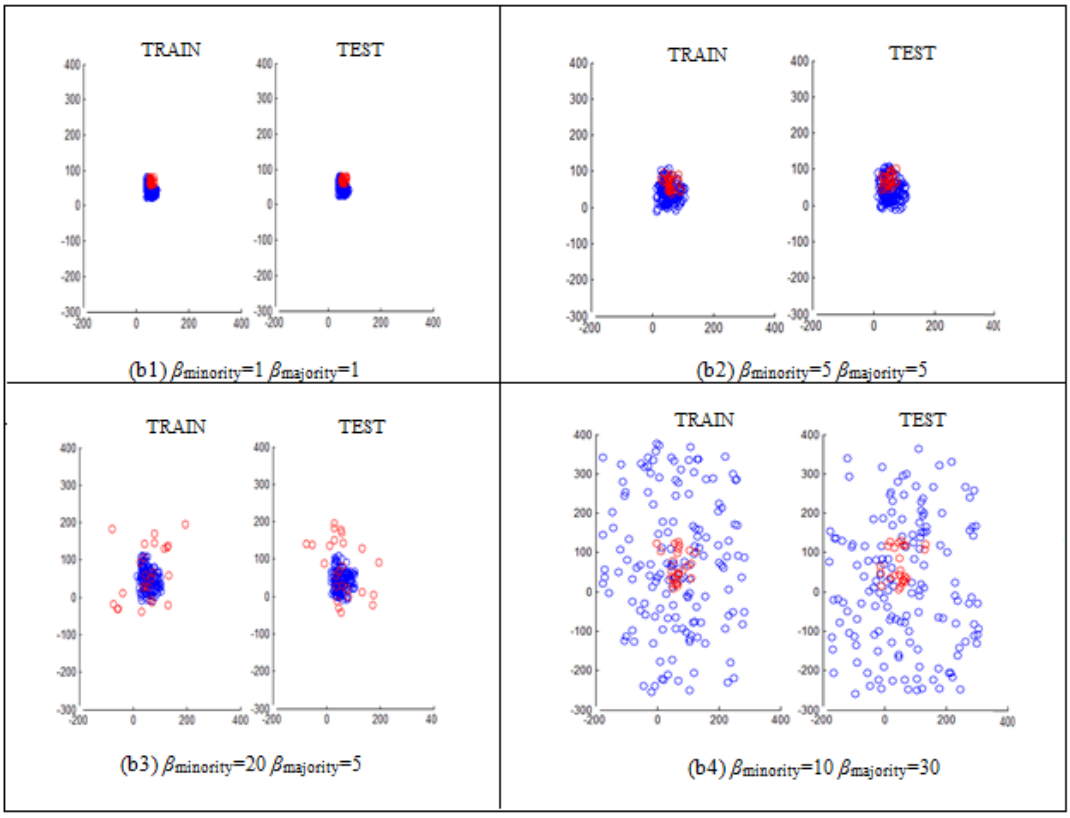
For the experiments with UMM; we randomly generated two overlapping uniform distributions for the minority and majority classes as the baseline data set. Then, similarly to the GMM while keeping the means of the distributions the same, we multiply the standard deviation with a constant $\beta = \{1, 2, 3, 4, 5, 6, 10, 20 \text{ and } 30\}$ and find the new intervals to generate random points. Increasing β makes the data set more overlapping compared to the baseline data set. In total 81 different data sets were obtained by using all pairs of β. For the baseline data set, β_{minority} and β_{majority} are equal to 1 while intervals of distributions are selected randomly. Generated GMM and UMM based synthetic data sets are available at http://homepages.inf.ed.ac.uk/s1064447/SytheticDataset_GMM_UMM.rar.

For all experiments in this section, Sequential Forward Feature Selection is applied to choose the best feature subset for fair comparison (since there is no prior information about features). Additionally, maximum achievable performance is also given (see below for the definition of GMM-OC). All the experiments were run with proposed method, SVM wFS, NB wFS, RF BT wFS, SVM SMOTE wFS, and NB SMOTE wFS with the settings given in Table 4.

In the experiments, the training, validation and test sets consist of 50%, 25% and 25% of the samples. The majority class samples and minority class samples were distributed appropriately to each set. All experiments were repeated 30 times with different data instances; therefore, for each fold the centers of the classes are also varied. Figure 5 shows examples of train and test data for the same set of class centers and for different values of α and β for GMM and UMM respectively.



(a) Distributions from GMM

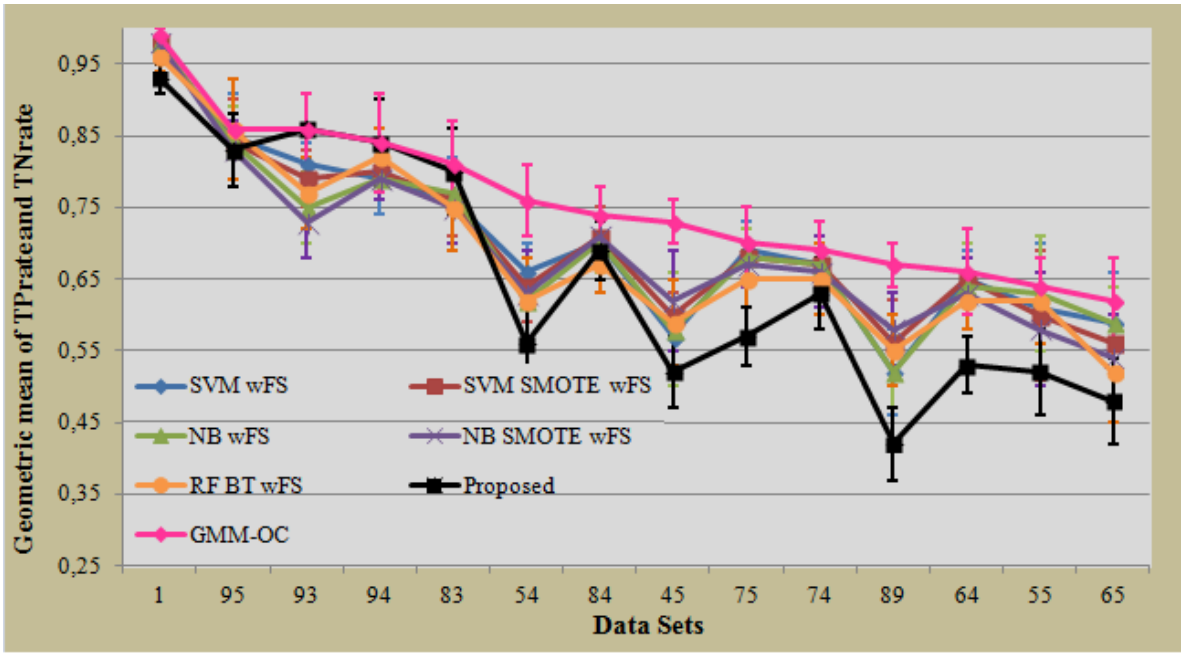


(b) Distributions from UMM

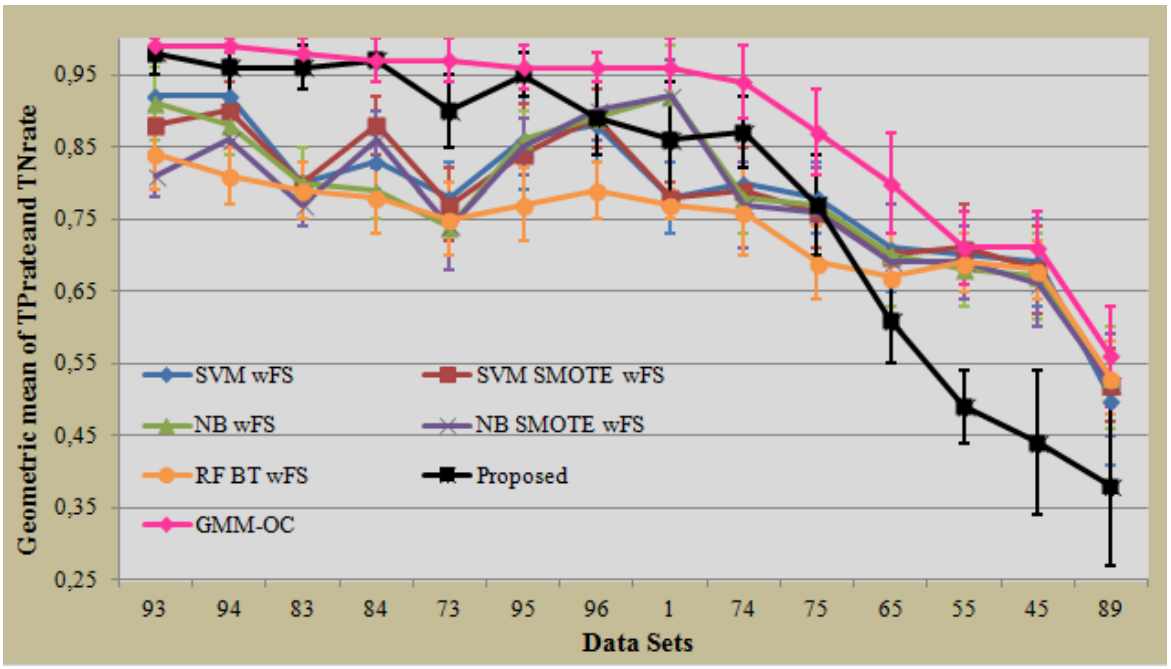
Fig. 5: Examples of train-test pairs when the number of features is 2. (a1-a4) GMM based data sets, (b1-b4) UMM based data sets with different α and β values for minority and majority classes for the same set of class centers. The class centers are varied on each cross validation fold.

The performance of the proposed hierarchical algorithm and comparison classifiers using the GMM based synthetic data sets (using 16 of the 100 pairs of α values which are enough to show the overall behavior of classifiers) with different numbers of features (2, 5 and 10) with 300 samples from the majority class and 50 samples from the minority class are shown in Figure 6. The GeoMean evaluation metric was used as it showed the worse performance (but still better than other methods) for the proposed method in Section 4.1.3.

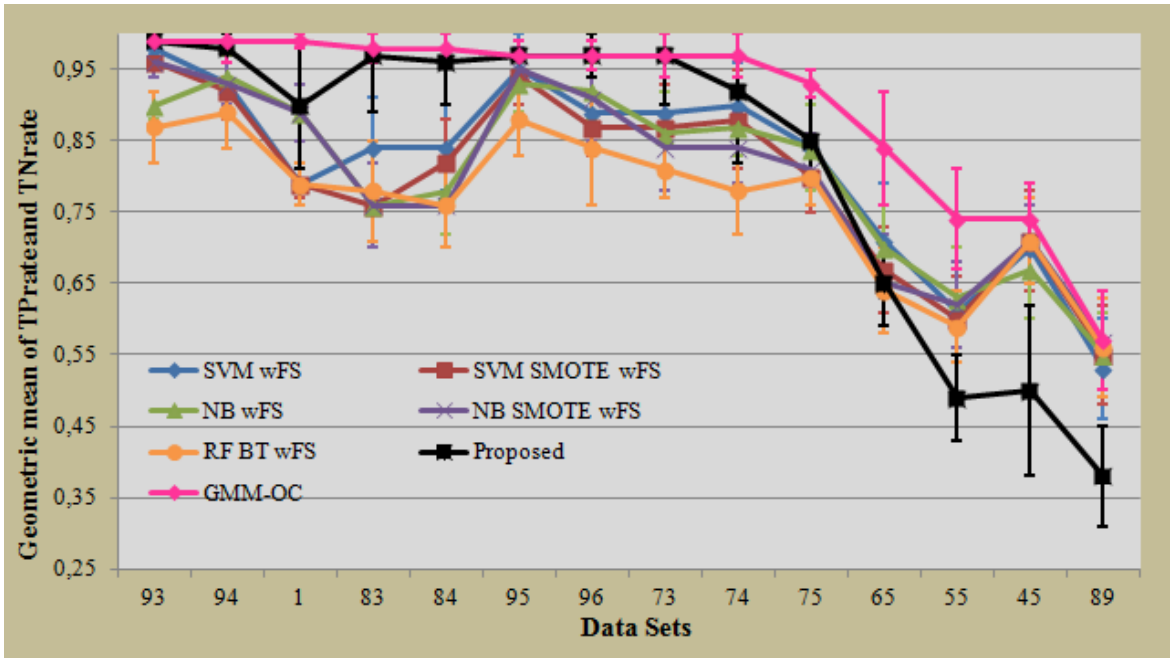
As well as classifying each sample according to the proposed method or comparison classifiers, we also calculated the posterior probabilities of test samples using the GMM that the corresponding data set was created from (including all training, validation and test samples) and classify each test sample according to the highest posterior probability while taking the prior probabilities equally. In other words, we calculated the class decisions using the GMM (that the data set is created from) itself as a classifier, which should model the asymptotic performance. We named this *GMM as optimal classifier* (**GMM-OC**) **which should show the maximum achievable performance for a given data set**. The optimal classifier helps us to understand how difficult it is to classify the data set. For instance, a low value of the geometric mean for GMM-OC means that the data distributions are overlapping. Conversely, high values mean separable classes whose classification should be easier. The results of GMM-OC are also given in Figure 6 while the results are sorted from best performance to worse performance of GMM-OC.



(a) Number of features is 2



(b) Number of features is 5



(c) Number of features is 10

Data set 1: $\alpha_{minority}$=1 $\alpha_{majority}$=1	**Data set 64:** $\alpha_{minority}$=64 $\alpha_{majority}$=8	**Data set 75:** $\alpha_{minority}$=128 $\alpha_{majority}$=16	**Data set 93:** $\alpha_{minority}$=512 $\alpha_{majority}$=4
Data set 45: $\alpha_{minority}$=16 $\alpha_{majority}$=16	**Data set 65:** $\alpha_{minority}$=64 $\alpha_{majority}$=16	**Data set 83:** $\alpha_{minority}$=256 $\alpha_{majority}$=4	**Data set 95:** $\alpha_{minority}$=512 $\alpha_{majority}$=16
Data set 54: $\alpha_{minority}$=32 $\alpha_{majority}$=8	**Data set 73:** $\alpha_{minority}$=128 $\alpha_{majority}$=4	**Data set 84:** $\alpha_{minority}$=256 $\alpha_{majority}$=8	**Data set 94:** $\alpha_{minority}$=512 $\alpha_{majority}$=8
Data set 55: $\alpha_{minority}$=32 $\alpha_{majority}$=16	**Data set 74:** $\alpha_{minority}$=128 $\alpha_{majority}$=8	**Data set 89:** $\alpha_{minority}$=256 $\alpha_{majority}$=256	**Data set 96:** $\alpha_{minority}$=512 $\alpha_{majority}$=32

Fig. 6: Results (given the best parameter setting) of methods on the GMM synthetic data: SVM wFS, NB wFS, RF BT wFS, SVM SMOTE wFS, NB SMOTE wFS, GMM-OC and the proposed hierarchical method in terms of the average of GeoMean for 16 different data sets created by different values of α for the minority and majority classes. The error bars show the standard deviation of the performance considering the 30 data folds for each data set. The number of features is a) 2, b) 5, c) 10, and the number of samples for the majority class is 300 while the number of samples in the minority class is 50. The data sets given on the horizontal axis refer to the index number of data set given in the legend which uses different combinations of $\alpha_{minority}$ and $\alpha_{majority}$ and are sorted by the performance of GMM-OC by decreasing order.

The performance (using GeoMean data) of the proposed method compared to other methods using 100 datasets for different numbers of features is summarized in Figure 7. In this Figure, different datasets are grouped by their α_{minority} and α_{majority} ratios (for instance α_{minority}=1, α_{majority}=1 and α_{minority}=512, α_{majority}=512 are put into same group). For each group, the number of experiments (each experiment consist of 30 data folds over the same dataset) that the proposed method performed better (better means if the proposed method had the best performance on average over all folds) is given over the total number of experiments for each group. The total performance of the proposed method over total number of datasets using each feature set is also given as "TOTAL". The results where proposed method performed better in the majority of the experiments is colored with light green. The results where proposed method performed worse than at least one other method in the majority of the experiments is colored with orange and the results where the proposed method performed about as well as the other methods (within ±0.02 of each other) are shown with pink color.

Based on the experimental results in Figure 6 and Figure 7, independently of the number of features, the performance of the proposed method increases when α_{minority} is larger than α_{majority}, (in other words if the minority class is sparser than the majority class) compared to data sets where the majority class is sparser than the minority or when they are equal. For the data sets where α_{minority} and α_{majority} are equal (or very close) and large enough (such as 128, 256) the minority class becomes inliers instead of being outliers. Therefore, the proposed method fails. On the other hand, when the number of features is increased even with a low ratio of α_{minority} to α_{majority}, the proposed method performs better (mostly the best). It performs similarly to other methods with a high value of $\alpha_{minority}$ and α_{majority} ratios, and it performs worse than the rest for small values of $\alpha_{minority}$ and α_{majority} ratios especially when the number of features is small such as 2. Considering all generated data sets, the proposed method performs better than the rest of the methods when the ratio of α_{minority} and α_{majority} is at least 32 with 2 features, at least 8 with 5 features and at least 0.0625 with 10 features. Those ratios (32, 8 and 0.0625) suggest that with more features, it is possible to have better performance by the proposed method even with low $\alpha_{minority}$ and α_{majority} ratios (which means less sparse minority class data).

$\alpha_{\text{minority}}/\alpha_{\text{majority}}$	2 features	5 features	10 features
1:512	0/1	0/1	0/1
1:256	0/2	0/2	0/2
1:128	0/3	0/3	0/3
1:64	0/4	0/4	1/4
1:32	0/5	0/5	2/5
1:16	0/6	0/6	3/6
1:8	0/7	0/7	4/7
1:4	0/8	2/8	4/8
1:2	0/9	3/9	5/9
1:1	1/10	4/10	5/10
2:1	3/9	4/9	5/9
4:1	2/8	3/8	4/8
8:1	2/7	4/7	4/7
16:1	2/6	4/6	5/6
32:1	3/5	4/5	5/5
64:1	3/4	4/4	4/4
128:1	3/3	3/3	3/3
256:1	2/2	2/2	2/2
512:1	1/1	1/1	1/1
TOTAL	22/100	38/100	57/100

Fig. 7: Summary of the performance (using GeoMean data) of the proposed method for different feature sets having 100 different datasets for each, grouped by $\alpha_{minority}$ and α_{majority} ratios. Color code: orange: proposed performs worse, green- proposed performs better, pink- about equal (see text for more detail).

Given that the proposed method potentially performs better than the rest of the methods when $\alpha_{minority} > \alpha_{majority}$, we also compared the methods with different imbalance ratios ($N_{minority}/N_{majority}$, where N represents the number of samples). Different imbalance ratios 0.67, 0.33 and 0.17 were used where the majority class has 300 samples and the minority class has 200, 100 and 50 samples respectively. We also varied the ratio of $\alpha_{minority}$ and $\alpha_{majority}$ as 32, 64 and 128. 30 trials were run for each experiment.

	$\alpha_{minority}/\alpha_{majority}$= 32	$\alpha_{minority}/\alpha_{majority}$= 64	$\alpha_{minority}/\alpha_{majority}$= 128
$N_{minority}/N_{majority}$=0.67	Proposed (+0.06)	Proposed (+0.06)	Proposed (+0.07)
$N_{minority}/N_{majority}$=0.33	All	Proposed (+0.03)	Proposed (+0.04)
$N_{minority}/N_{majority}$=0.17	All	Proposed (+0.03)	Proposed (+0.05)

(a) Number of features is 2

	$\alpha_{minority}/\alpha_{majority}$= 32	$\alpha_{minority}/\alpha_{majority}$= 64	$\alpha_{minority}/\alpha_{majority}$= 128
$N_{minority}/N_{majority}$=0.67	SVM SMOTE wFS NB SMOTE wFS (-0.06)	SVM SMOTE wFS NB SMOTE wFS (-0.06)	All
$N_{minority}/N_{majority}$=0.33	All	All	All
$N_{minority}/N_{majority}$=0.17	Proposed (+0.09)	Proposed (+0.16)	Proposed (+0.06)

(a) Number of features is 5

	$\alpha_{minority}/\alpha_{majority}$= 32	$\alpha_{minority}/\alpha_{majority}$= 64	$\alpha_{minority}/\alpha_{majority}$= 128
$N_{minority}/N_{majority}$ =0.67	All	All	All
$N_{minority}/N_{majority}$=0.33	All	All	All
$N_{minority}/N_{majority}$=0.17	Proposed (+0.12)	Proposed (+0.13)	All

(a) Number of features is 10

Fig. 8: Best classification performance of methods (SVM wFS, NB wFS, RF BT wFS, SVM SMOTE wFS, and NB SMOTE wFS, and the proposed hierarchical decomposition method) in terms the average GeoMean using data sets with different imbalanced ratios ($N_{minority}/N_{majority}$) when $\alpha_{minority}$, $\alpha_{majority}$ is 32, 64 and 128. "All" means that all classifiers had essentially equal performance.

Figure 8 shows similar performances by classifiers (i.e. when within ±0.02 of each other) as "All". For 2 features, in "All" cases, the classifiers achieved approximately 0.70 (average geometric mean of sensitivity and specificity for 2 folds, repeated 30 times). For 5 and 10 features, "All" refers to performances over 0.96. For the cases when the proposed method performed substantially better than the rest we stated how much better it performed compared to the next best performance by giving the difference after "+" sign. For instance, for 10 features when the $N_{minority}/N_{majority}$=0.17 and $\alpha_{minority}/\alpha_{majority}$= 32 the proposed method's performance is 0.96 while the next best performance is by SVM SMOTE wFS with 0.84. On the other hand, if the performance of the proposed method is worse than any method then we state the best classifier with its name and the performance difference with the proposed method after "-" sign. For example, for 5 features, $N_{minority}/N_{majority}$ =0.67 and $\alpha_{minority}/\alpha_{majority}$= 32 SVM SMOTE wFS and NB SMOTE wFS performed the best with 0.96 while the proposed method performed 0.90. The proposed method performs better as the number of features is increased (for instance for $N_{minority}/N_{majority}$=0.67 and $\alpha_{minority}/\alpha_{majority}$= 32; it performed 0.83 with 2 features, 0.90 with 5 features and 0.97 with 10 features). The proposed method performs better than the other methods when the imbalanced ratio is low.

We observed similar results when using UMM. The results with UMM also showed that the proposed method performs better compared to the other methods when $\beta_{minority}$ is larger than β_{majority}, when the number of features becomes larger, and the imbalanced ratio of majority and minority classes gets larger. We do not add these results to this paper as they do not make much further contribution to understanding the behavior of the proposed method. But it shows that the proposed method is at least somewhat independent to the distribution of the data.

5. Discussions and Conclusions

We presented a hierarchical decomposition method which is based on clustering and uses outlier detection as the classifier. The hierarchy is based on the similarities of data (ie. clusters) where the hierarchy levels are built using different data and feature subsets. The key observation and the justification for using a hierarchy is that some features allow partitioning of some samples, which then allows other features to be useful on the remaining samples. Outlier detection was used to detect minority class samples assuming that the minority class samples in each class are outliers by cardinality or by their distance to class center.

The proposed method comes up with multiple decision boundaries which are equal to the number of clusters in a hierarchy level (see Figure 9). Those boundaries help the classification of data especially if the data is highly overlapping, and the imbalanced ratio between minority and majority class is high.

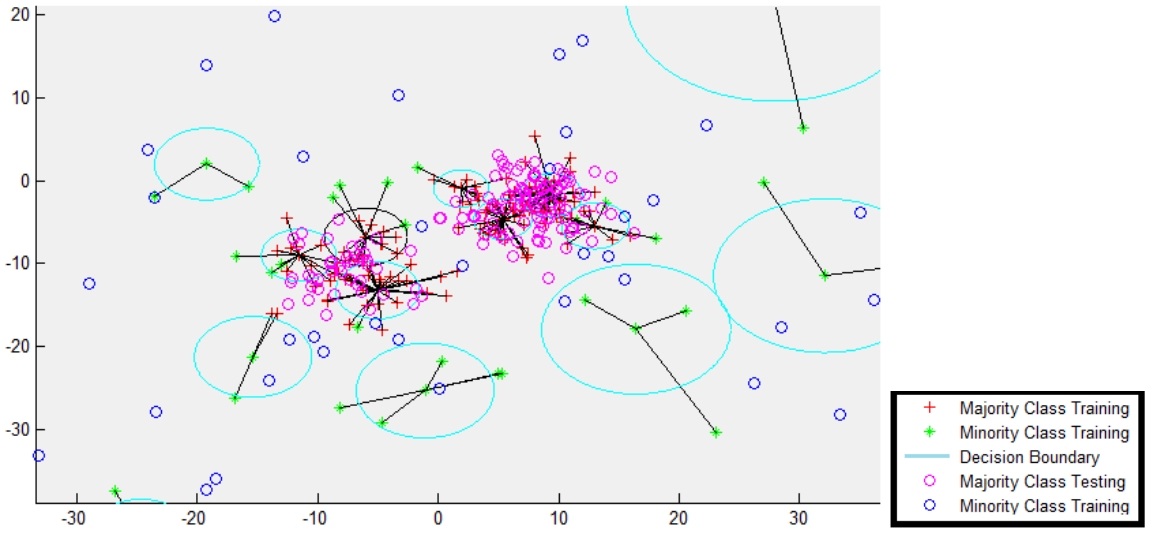


Fig. 9: A plot of outlier boundaries (light blue) for training data with superimposed testing data. This figure illustrates the first level of the hierarchy (including misclassified clusters) for a data set having 5 features where only 2 features were selected at that level.

As seen from the results, the proposed method does not need the support of any cost function, algorithmic or data level algorithm (see Section 1 for definitions) to handle imbalanced data sets. On the other hand, since it does not use all the data samples to build up the hierarchy at each level, it can be considered close to bagging. In our case, the bags are defined by the performance of the classifier (we continue to build up the hierarchy with the incorrectly classified samples) but not as random

subsets as happens in bagging. Moreover, it is different from boosting by using a subset of data in addition to not using a weight to support the classification of misclassified samples.

The results showed that the proposed method's majority class classification performance (Specificity, TNrate) is better than its minority class classification performance (Sensitivity, TPrate) in overall (the results in terms of AGeoMean supports this). The main reason for this is the heuristic that we used to classify the new data sample. In this context, a class decision as the majority class at any level of the hierarchy stops the classification of the new data while a decision as the minority class sends it to the next hierarchy level (see Section 3.5 for more detail). The heuristic results in a higher TNrate with lower TPrate compared to use the opposite heuristic (classification of the data as the minority class stops the classification whereas classification of the data as the majority class sends the data to the next hierarchy level). The heuristic used also has a lower GeoMean compared to the opposite heuristic since any increase in TPrate contributes to the GeoMean more than any increase in TNrate (For instance, TNrate=80/100, TPrate=30/50 makes the GeoMean= 0.69, whereas TNrate=70/100, TPrate=40/50 makes the GeoMean=0.75). Therefore, when the GeoMean is used as the evaluation function, the opposite heuristic (which favors the minority class) will potentially have a better evaluation score than the heuristic used here (that favors the majority class) on the same data set. Those claims are valid for AUC as an increase in the TNrate means a decrease in the FPrate which means better AUC performance. Considering this, we can say that the proposed method already performs well while it is possible to increase its performance in terms of GeoMean and AUC by using the opposite heuristic. Additionally, the opposite heuristic might be useful in some applications such as anomaly detection where the classification of the minority class is more important than the classification of the majority class.

The computational complexity during training of the proposed method is much more than that of the other methods which can be seen as a shortcoming. To decrease the training time complexity, feature selection can be implemented in parallel on a task farming architecture with the methodology given in [72]. However, more importantly, the proposed method's testing complexity is as efficient as the other methods, which are simply a few distance calculations between the closest clusters at each level and the new data point.

In conclusion, the proposed hierarchical decomposition method is successful at classifying imbalanced data sets even though the majority and minority classes contain varieties, and classes overlap (frequently seen in real life applications). It performs much better if the minority samples are sparse compared to the majority samples where popular classification methods generally fail. It also performs well when the ratio between minority and majority samples is low.

The future work is to test the performance of the proposed hierarchical method on specific real–life problems such as anomaly detection in video surveillance. Additionally, a comparative analysis in terms of the possible heuristics for new data classification and different outlier detection strategies can be performed. The scalability of the proposed method should also be evaluated on very large data sets.

Acknowledgements

We thank Bas Boom and Steven McDonagh for helpful discussion. We also thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. The first author of this work is funded by University of Edinburgh and School of Informatics.

References

[1] K. N. Rao, T. V. Rao, D. R. Lakshmi, A Novel Class Imbalance Learning Method using Subset Filtering, International Journal of Scientific and Engineering Research 3 (9) (2012) 1-9.

[2] M. Galar, A. Fernandez, E. Barrenechea, F. Herrera, EUSBoost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling, Pattern Recognition 46 (2013) 3460-71.

[3] V. C. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, SMOTE: Synthetic Minority Over-sampling technique, Journal of Artificial Intelligence Research 16 (2002) 321-357.

[4] S. García, F. Herrera, Evolutionary Undersampling for Classification with imbalanced datasets: proposals and taxonomy, Evolutionary Computation 17 (2009) 275–306.

[5] Y. Tang, Y. Q. Zhang, N. V. Chawla, S. Krasser, SVMs modeling for highly imbalanced classification, IEEE Transactions on Systems Man. Cybernetics Part B 39 (1) (2009) 281–288.

[6] B. Das, N. C. Krishnan, D. J. Cook, Handling Imbalanced and Overlapping Classes in Smart Environments Prompting Dataset, Springer Book on Data Mining for Services in Studies in Computational Intelligence (2012).

[7] N. Japkowicz, S. Stephen, The class imbalance problem: a systematic study, Intelligent Data Analysis 6 (5) (2002) 429–449.

[8] M. Kubat, S. Matwin, Addressing the curse of imbalanced training sets: one-sided selection, in: Proceedings of the 14th International Conference on Machine Learning (ICML'97), 179–186, 1997.

[9] R. Batuwita, V. Palade, Adjusted Geometric-mean: A Novel Performance Measure for Imbalanced Bioinformatics Dataset Learning, Journal of Bioinformatics and Computational Biology 10(4) (2012) 1-23.

[10] J. Huang, C. X. Ling, Using AUC and accuracy in evaluating learning algorithms, IEEE Transactions on Knowledge and Data Engineering 17 (3) (2005) 299–310.

[11] S. Vajda, G. A. Fink, Strategies for Training Robust Neural Network Based Digit Recognizers on Unbalanced Data Set, in: Proceedings of the 12th Conference on Frontiers in Handwriting Recognition 148-153 2010.

[12] G. Wu, E. Y. Chang, Class-Boundary Alignment for Imbalanced Dataset Learning. in: Proceedings of International Conference on Machine Learning 2003 Workshop on Learning from Imbalanced Data Sets II, Washington, DC, 2003.

[13] M. Pazzani, C. Merz, P. Murphy, K. Ali, T. Hume, C. Brunk, Reducing Misclassification Costs, in: Proceedings of the 11th International Conference on Machine Learning 217-225 1994.

[14] P. Domingos, Metacost: A General Method for Making Classifiers Cost-sensitive, in: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 155–164 1999.

[15] N. Japkowicz, The Class Imbalance Problem: Significance and Strategies, in: Proceedings of the 2000 International Conference on Artificial Intelligence, Special Track on Inductive Learning, 2000.

[16] C. Ling, C. Li, Data Mining for Direct Marketing Problems and Solutions, in: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining 1998.

[17] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince and F. Herrera, A review on Ensembles for the Class Imbalance Problem: Bagging, Boosting and Hybrid-Based Approaches, IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews, 42(4) (2012) 463-484.

[18] L. Breiman, Bagging predictors, Machine Learning 24 (1996) 123–140.

[19] Y. Freund, R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences, 55(1) (1997) 119–139.

[20] N. V. Chawla, A. Lazarevic, L. O. Hall, K. W. Bowyer. Smoteboost: Improving prediction of the minority class in boosting, in: Proceedings of the Seventh European Conference on Principles and Practice of Knowledge Discovery in Databases 107-119 2003.

[21] B. Epshtein, S. Ullman, Feature hierarchies for object classification, in: Proceedings of International Conference on Computer Vision (ICCV), 220-227, 2005.

[22] Y. Chen, M. M. Crawford, J. Ghosh, Integrating support vector machines in a hierarchical output space decomposition framework, in: Proceedings of the IEEE International Symposium on Geoscience and Remote Sensing 2 (2004) 949-952.

[23] C. Chen, A. Liaw, L. Breiman, Using random forest to learn imbalanced data, Technical Report 666, Statistics Department, University of California at Berkeley, Available at http://www.stat.berkeley.edu/users/chenchao/666.pdf 2004.

[24] R. A. Johnson, N. V. Chawla, J. J. Hellman, Species Distribution Modeling and Prediction: A Class Imbalance Problem, in: Proceedings of Intelligent Data Understanding (CIDU), 9-16, 2012.

[25] H. Han, W. Y. Wang, B. H. Mao, Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning, Advances in Intelligent Computing (2005) 878-887.

[26] S. Hu, Y. Liang, L. Ma, Y. He, MSMOTE: Improving classification performance when training data is imbalanced, in: Proceedings of 2nd International Workshop on Computer Science Engineering. 2, 13–17, 2009.

[27] S. Barua, M. Islam, X. Yao, and K. Murase, MWMOTE-Majority Weighted Minority Oversampling Technique for Imbalanced Data Set Learning, IEEE Transactions on Knowledge and Data Engineering, 26(2) (2014) 405-425.

[28] P. Radivojac, N. V. Chawla, K. Dunker, Z. Obradovic, Classification and Knowledge Discovery in Protein Databases. Journal of Biomedical Informatics 37 (4) (2004) 224-239.

[29] X. Y. Liu, J. Wu, Z. H. Zhou, Exploratory undersampling for class imbalance learning, IEEE Transactions on Systems Man. Cybernetics Part B 39 (2) (2009) 539–550.

[30] C. Seiffert, T. M. Khoshgoftaar, J. Hulse, A. Napolitano, RUSBoost: A Hybrid Approach to Alleviating Class Imbalance, IEEE Trans. On Systems Man and Cybernetics-Part A: Systems and Humans 20(1) (2010).

[31] B. X. Wang, N. Japkowicz, Boosting Support Vector Machines for Imbalanced Data Sets, Lecture Notes in Artificial Intelligence 4994 38-47 2008.

[32] S. Tan, Neighbor-weighted k-nearest neighbor for unbalanced text corpus, Expert System Applications, 28(4) (2005) 667–671.

[33] G. Fumera, F. Roli, Cost-sensitive Learning in Support Vector Machines, in: Proceedings of Workshop Machine Learning, Methods and Applications, held in the Context of the 8th meeting of the Italian Assoc. of Artificial Intelligence (2002).

[34] C. Drummond, R. C. Holte. Exploiting the cost (in)sensitivity of decision tree splitting criteria, in: Proceedings of the Seventeenth International Conference on Machine Learning (ICML) 239–246, 2000.

[35] D. P. Williams, V. Myers, M. S. Silvious, Mine Classification With Imbalanced Data, IEEE Geosciences And Remote Sensing Letters, 6(3), 2009.

[36] S. Kotsiantis, D. Kanellopoulos, P. Pintelas, Handling unbalanced datasets: A review, GESTS International Transactions on Computer Science and Engineering 30 (1) (2006) 25-36.

[37] R. Longadge, S. S. Dongre, L. Malik, Class Imbalance Problem in Data Mining: Review, International Journal of Computer Science and Network (IJCSN) 2 (1) (2013).

[38] V. Lopez, A. Fernandez, S. Garcia, V. Palade, F. Herrera, An insight into classification with imbalanced data: Empirical results and current trends on using intrinsic characteristics, Information Sciences 150 (2013) 113-141.

[39] H. He, E. A. Garcia, Learning from Imbalanced Data, IEEE Trans. on Knowledge and Data Engineering 21(9) (2009) 1263-1284.

[40] C.N. Silla, A. A. Freitas, A survey of hierarchical classification across different application domains, Data Mining and Knowledge Discovery 22(1-2) (2010) 31–72.

[41] N. Cesa-Bianchi, C. Gentile, L. Zaniboni, Incremental algorithms for hierarchical classification, The Journal of Machine Learning Research 7 (2006) 31-54.

[42] F. Wu, J. Zhang, V. Honavar, Learning classifiers using hierarchically structured class taxonomies, in: Proceedings of the Symposium on Abstraction, Reformulation, and Approximation, Springer 3607 313-320, 2005.

[43] T. Li, M. Ogihara, Music genre classification with taxonomy, in: Proceedings of IEEE Int. Conf. on Acoustics Speech and Signal Processing 197-200 2005.

[44] P. N. Bennett, N. Nguyen, Refined experts: improving classification in large taxonomies, in: Proceedings of the 32nd International ACM SIGIR conference on Research and development in information retrieval 11-18 2009.

[45] C. N. Silla Jr., A. A. Freitas, Selecting different protein representations and classification algorithms in hierarchical protein function prediction, Intelligent Data Analysis Journal, 15(6) 979-999 2011.

[46] S. Kumar, J. Ghosh, M. M. Crawford, Hierarchical fusion of multiple classifiers for hyperspectral data analysis, Pattern Analysis and Applications 5 (2002) 210-220.

[47] P. Y. Hao, J. H. Chiang, Y. K. Tu, Hierarchically SVM classification based on support vector clustering method and its application to document categorization, Expert Systems with Applications 33 (2007) 627-635.

[48] C. O. A. Freitas, L. S. Oliveira, S. B. K. Aires, F. Bortolozzi, Metaclasses and zoning mechanism applied to handwriting recognition, Journal of Universal Computer Science 14(2) (2008) 211–223.

[49] Y. Peng, C. Lin, M. Sun, Audio classification using binary hierarchical classifiers with feature selection for healthcare applications, in: Proceedings of IEEE International Symposium on Circuits and Systems 3238-3241 2008.

[50] P. X. Huang, B. J. Boom, B. F. Fisher, Underwater Live Fish Recognition using a Balanced-Guaranteed Optimized Tree, in: Proceedings of 11th Asian Conference on Computer Vision (ACCV), 2012.

[51] C. Freeman, D. Kulic, O. Basir, Joint feature selection and hierarchical classifier design, IEEE International Conference on Systems Man and Cybernetic (2011) 1728-1734.

[52] B. J. Frey, D. Dueck, Clustering by Passing Messages Between Data Points, Science (2007) 972-97.

[53] N. Anjum, A. Cavallaro, Multifeature Object Trajectory Clustering for Video Analysis, IEEE Transaction on Circuits Systems Video Technology 18(11) (2008) 1555-1564.

[54] P. Pudil, J. Novovicova, J. Kittler, Floating Search Methods in Feature Selection, Pattern Recognition Letters 15(11) 1119-1125 1994.

[55] X. He, D. Cai, P. Niyogi, Laplacian score for feature selection, In Advances in Neural Information Processing Systems 18 (2005).

[56] D. Cai, C. Zhang, X. He, Unsupervised feature selection for multi-cluster data, in: Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining 333-342 2010.

[57] D. Yao, J. Yang, Xiaojuan Zhan, An Improved Random Forest Algorithm for Class-Imbalanced Data Classification and its Application in PAD Risk Factors Analysis, The Open Electrical & Electronic Engineering Journal 7 (2013) 62-70.

[58] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, KEEL data-mining software tool: dataset repository, integration of algorithms and experimental analysis framework, Journal of Multiple-Valued Logic and Soft Computing, 17(2-3) (2011) 255-287. http://www.keel.es/dataset.php

[59] V. G. Sigillito, S. P. Wing, L. V. Hutton, K. B. Baker, Classification of radar returns from the ionosphere using neural networks, Johns Hopkins APL Technical Digest. 10 (1989) 262–266.

[60] C. L Blake., C. J. Merz, UCI repository of machine learning databases, 1998. http://archive.ics.uci.edu/ml/ Accessed Oct 28, 2013.

[61] Y. Hayashi, Neural expert system using fuzzy teaching input and its application to medical diagnosis, Information Sciences Applications, 1 (1994) 47-58.

[62] J. W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler, R. S. Johannes, Using the ADAP learning algorithm to forecast the onset of diabetes mellitus, in: Proceedings of the Symposium on Computer Applications and Medical Care 261–265 1988.

[63] S. Weiss, I. Kapouleas, An empirical comparison of pattern recognition. neural nets and machine learning classification methods, in: Proceedings of the 11th International Joint Conference of Artificial Intelligence 781:787 1989.

[64] M. R. Boutell, J. Luo, X. Shen, C. M. Brown, Learning multi-label scene classification. Pattern Recognition 37(9) (2004) 1757-1771.

[65] M. Kubat, R. Holte, S. Matwin, Machine learning for the detection of oil spills in satellite radar images, Machine Learning 30 (1998) 195-215.

[66] J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, 1993.

[67] L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone, Classification and Regression Trees, Belmont CA: Wadsworth 1984.

[68] S. García, F. Herrera, An extension on statistical comparisons of classifiers over multiple data sets for all Pairwise Comparisons, Journal of Machine Learning Research 9 (2008) 2677-2694.

[69] S. Garcia, A. Fernandez, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental Analysis of Power. Information Sciences 180 (2010) 2044-2064.

[70] J. Demsar, Statistical comparisons of classifiers over multiple datasets, Journal of Machine Learning Research 7 (2006) 1-30.

[71] S. Holm, A simples equentially rejective multiple test procedure, Scandinavian Journal of Statistics 6 (1979) 65–70.

[72] S. McDonagh, C. Beyan, P. X. Huang, R. B. Fisher, Applying semi-synchronised task farming to large-scale computer vision problems, The International Journal of High Performance Computing Applications (2014) 1-24.